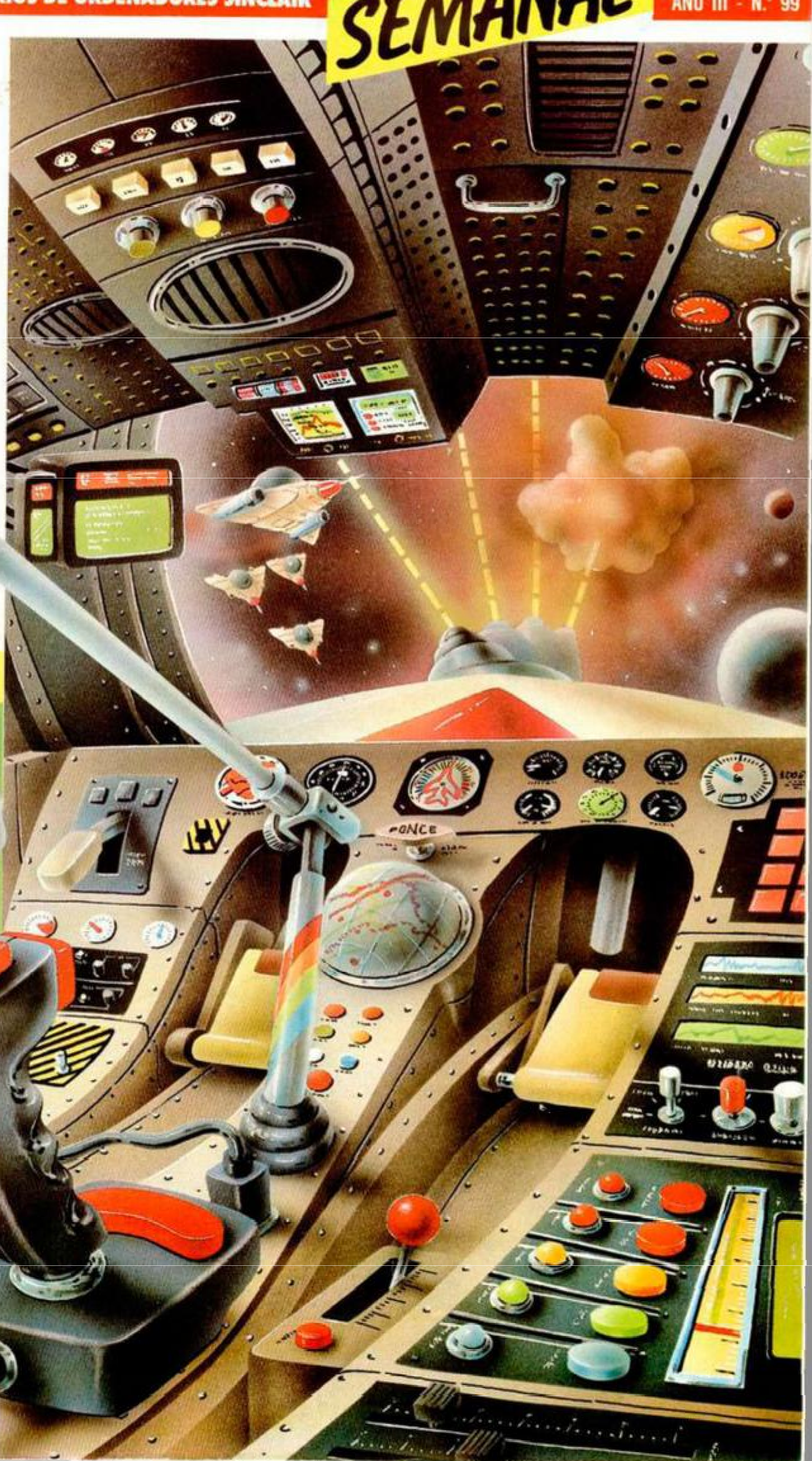
AÑO III - NUM. 99

MICROHOBBY SEMANAL

REVISTA INDEPENDIENTE PARA USUARIOS DE ORDENADORES SINCLAIR

AÑO III - N.º 99

135 PTS.
Canarias 140 ptas.



NUEVO

"BOBBY BEARING" EL DESAFÍO DEL PLANETA DE ACERO

MICROMANÍA

CONSIGUE VIDAS INFINITAS EN "EQUINOX" "DYNAMITE DAN" Y "RASPUTÍN"

HARDWARE

INSTALA EN TU JOYSTICK UN MECANISMO DE DISPARO AUTOMÁ-TICO

PROGRAMACIÓN

LAS RUTINAS DE MÚSICA PARA EL "CÓMEME"

¡Con notas simultáneas!

HOBBY PRESS

¡¡Gratis!!

Suscríbete a Microhobby **o realiza ahora tu renovación y recibirás,** *totalmente gratis,* **este magnífico regalo.**

Kit profesional de ajuste y mantenimiento.



Contiene:
— Destornillador especial para ajuste de azimuth
— Spray limpiador de cabezas magnéticas «Computer Cleaner»
— Cassette con instrucciones de uso grabadas

Envíanos hoy mismo el cupón de suscripción que se encuentra cosido en las páginas de esta revista y te asegurarás todo un año de lectura estimulante y, además, evitarás todos tus problemas de carga.

¡PON A PUNTO TU CASSETTE Y OLVIDATE DE LOS PROBLEMAS DE CARGA!

(Oferta válida sólo para España, hasta el 31 de octubre de 1986).

MICROHOBBY ESTA SEMANA

AÑO III. N.º 99. 21 al 27 de octubre de 1986.
135 ptas. Canarias, Ceuta y Melilla: 130 ptas.
Sobretasa aérea para Canarias: 10 ptas.

4 MICROPANORAMA.

7 COMIC. La Armadura Sagrada de Antiriad (y IV)

10 PROGRAMAS MICROHOBBY. «Solitario del 7».

14 NUEVO. Bobby Bearing. Knight Rider. Nightmare Rally. Strike Force Cobra.

18 PROGRAMACION. Aprende a programar tu propio juego. «Cómeme III».

22 MICROMANIA / LIBRO.

23 JUSTICIEROS DEL SOFTWARE. «Las tres luces de Glaurung».

24 HARDWARE. Las interfaces y autodisparo en el joystick.

28 INICIACION. El sistema «Filmation» paso a paso (III).

31 TRUCOS.

32 CONSULTORIO.

34 OCASION.





El autodisparo en los joysticks.
(Pág. 18)

Director Editorial
José I. Gómez-Centurión

Director Ejecutivo
Domingo Gómez

Asesor Editorial
Gabriel Nieto

Diseño
Cristina Gómez

Redacción
Amalio Gómez, Pedro Pérez, Jesús Alonso

Secretaría Redacción
Carmen Santamaría

Colaboradores
Primitivo de Francisco, Rafael Prades, Miguel Sepúlveda, Sergio Martínez, J. M. Lazo, Paco Martín

Corresponsal en Londres
Alan Heap

Fotografía
Carlos Candel
Chema Sacristán

Portada
José María Ponce

Dibujos
Teo Mójica, F. L. Frontán, J. M. López Moreno, J. Igual, J. A. Calvo, Lóriga, J. Olivares

Edita
HOBBY PRESS, S. A.

Presidente
María Andrino

Consejero Delegado
José I. Gómez-Centurión

Jefe de Producción
Carlos Peropadre

Publicidad
Mar Lumbreras

Secretaría de Dirección
Pilar Aristizábal

Suscripciones
M.ª Rosa González
M.ª del Mar Calzada

Redacción, Administración y Publicidad
Ctra. de Irún Km 12,400
28049 Madrid
Tel: 734 70 12
Télex: 49480 HOPR

Pedidos y Subcripciones
Tel: 734 65 00

Dto. Circulación
Paulino Blanco

Distribución
Coedis, S. A. Valencia, 245
Barcelona

Imprime
Rotedic, S. A. Ctra. de Irún, km 12,450 (MADRID)

Fotocomposición
Novocomp, S.A.
Nicolás Morales, 38-40

Fotomecánica
Grof
Ezequiel Solana, 16

Depósito Legal
M-36.598-1984

Representante para Argentina, Chile, Uruguay y Paraguay, Cía. Americana de Ediciones, S.R.L. Sud América 1.532. Tel.: 21 24 64. 1209 BUENOS AIRES (Argentina)

Solicitado control OJD

MICROPANORAMA

TODA UNA ORQUESTA EN EL RAM MUSIC MACHINE



El RAM Music Machine es algo completamente nuevo para los ordenadores personales. Por primera vez, el poseedor de un Spectrum podrá crear sonidos, componer música, añadir percusiones e incluso controlar otros instrumentos musicales, todo ello gracias a un sistema único y sencillo de manejar. Por sus pantallas de menú, incluso los principiantes pueden comenzar directamente a crear sus canciones.

Music Machine es un periférico compacto, el cual, conectado al bus de expansión del ordenador y a un equipo de alta fidelidad ,puede actuar incluso como un Midi. Las posibilidades sonoras de este aparato son verdaderamente sorprendentes, y de entre sus circuitos salen los sonidos más realistas, haciendo olvidar por completo el típico sonido de chip.

Los menús de pantallas incluyen muy variadas opciones, las cuales van desde la representación gráfica de los sonidos, composición de secuencias realizadas por teclado o percusión, realización de partituras, control y manejo de otros instrumentos a través del Midi, y una considerable oferta de posibilidades de ejcución directa o programada.

El sistema se acompaña de una cinta de demostración, la cual confirma las increíbles posibilidades de este Music Machine.

AMX MOUSE

IMAGINACION EN LA PUNTA DE TUS DEDOS

Un ratón, más que un periférico, es casi una ventana abierta a la fantasía y al arte. El propio teclado o cualquier joystick pueden emular estas mismas funciones de utensilios de diseño, pero el ratón es, sin duda, la mejor herramienta que se puede utilizar a la hora de plasmar nuestro arte en la pantalla de un ordenador.

AMX es una de las compañías que más intensamente está trabajando en este campo, pues varios son ya los modelos que posee en el mercado, pero ésta su última creación supera en calidad y prestaciones a todas las anteriores.

El lote inicial incluye, además del propio ratón, que dispone de tres botones de control, el software artístico más sofisticado compuesto por ventanas, iconos, menús y punteros. Su calidad de trabajo a la hora de realizar, imprimir o grabar los diseños, es sorprendente. También incluye el software necesario para modificar según las necesidades personales de cada usuario el desarrollo y funciones del ratón a través de Basic.

El AMX Mouse es compatible con toda la gama de Amstrad, incluso los PC, con BBC y con los Spectrum 48, 48 + , y 128. Por otra parte puede ser utilizado para controlar los mejores programas de diseño como puede ser el Art, Colour Palette, Control o Mouse Mat de AMX, o el Art Studio de Rainbird, así como los programas de Softechnics, Artist II y Writer.

AQUI LONDRES

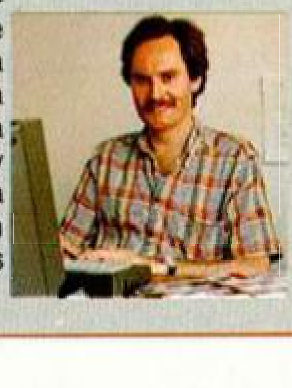
Las máquinas de videojuegos se han vuelto a poner de moda. La compañía de juguetes Mattel ha anunciado que va a lanzar en el Reino Unido, la máquina de videojuegos japoneses «Nigtendo». Esta noticia ha sido publicada y confirmada tras una comunicación de Ariola Soft en la que se decía que esta compañía tenía intenciones de importar al Reino Unido el videoconsola de juegos de «Sega» para las próximas navidades.

«Memotech», a su vez, se encuentra desarrollando un nuevo microordenador que llevará por nombre «Mtx-Marck 2». Este nuevo modelo, incluye un sistema para cassette incorporado, una memoria Ram de 256 K y una Ram Vídeo de 16 K, además de 6 juegos. El precio oscilará alrededor de 100 libras sin incluir el cassette y el monitor.

«Code Masters», una nueva marca comercial, se ha dado a conocer recientemente. Dicha firma está compuesta por un padre de familia y sus dos hijos, quienes han sido los autores de programas de Mastertronic como son «The Last V8« y «Masters of Magic».

Su intención es publicar 50 títulos para estas navidades y hacerse con el 10 por 100 del mercado de Inglaterra, en un período no superior a un año.

Amstrad se está diversificando en el terreno de la televisión vía satélite. El director de Amstrad, Alan Suggar, ha afirmado su intención de producir una antena para TV satélite a 99 libras y que saldrá a la venta en los próximos años 90.



ASTERIX Y EL CALDERO MAGICO: DE LAS GALIAS AL ORDENADOR

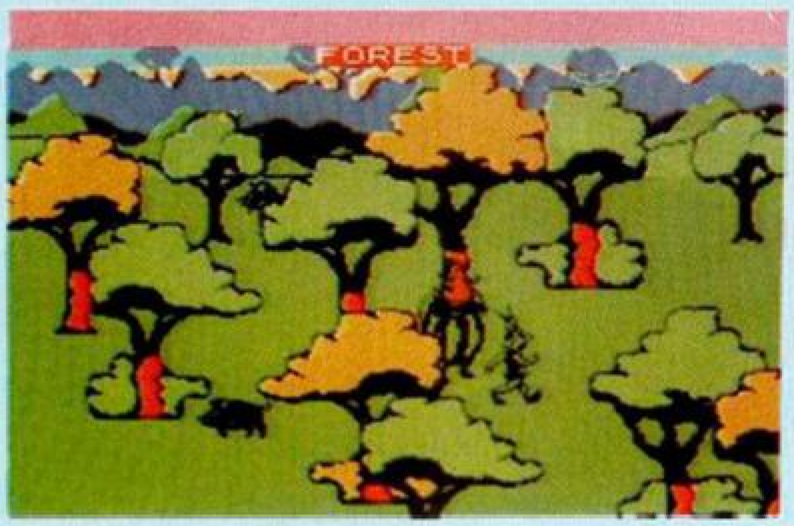
Asterix, una de las más brillantes estrellas pertenecientes al mundo del cómic y el cine, acaba de hacer su presentación en un programa para ordenador. La realización de dicho programa ha corrido a cargo de la prestigiosa compañía Merbourne House, quienes han obtenido como resultado un juego de una calidad excelente, tanto en lo relativo a sus aspectos de divertimento como de diseño.

La vistosidad de cada una de las numerosas pantallas que configuran el programa es sorprendente. Hasta el momento se han realizado las dos versiones correspondientes para Spectrum y Commodore, y en ambos los resultados han sido semejantes. En Commodore no es demasiado de extrañar que los gráficos sean de una buena calidad debido a las mayores posibilidades que ofrecen sus 64 K de memoria, pero sí es un poco más extraño que los gráficos para Spectrum hayan quedado tan similares disponiendo de unos medios considerablemente menores de capacidad.

Este demuestra una gran labor de programación y una notable habilidad para aprovechar las posibilidades de un ordenador no demasiado potente.

Asterix y el Caldero Mágico, es un juego planteado como arcade. En él, el pequeño galo acompañado de su inseparable amigo Obelix deberá recoger las porciones del caldero en el cual el druida del poblado fabricaba la porción mágica. Como sabréis ésta es una bebida indispensable para ellos, pues gracias a sus increíbles efectos, los miembros de esta pequeña aldea pueden defenderse de los intentos de conquista por parte de sus vecinos, los romanos.

Tan honorable personaje no podía haber entrado en el ordenador con mejor pie, gracias a la habilidad de estos extraordinarios programadores, australianos para más señas.



20 +

CLASIFICACION	SEMANAS PREM.	TENDENCIA		SPECTRUM	AMSTRAD	COMMODORE	MSX
1	15	–	**GREEN BERET.** Imagine.	●	●	●	
2	4	↑	**KUNG-FU MASTER.** US Gold	●			●
3	6	–	**LAS TRES LUCES DE GLAURUNG.** Erbe	●	●		
4	15	↑	**THE WAY OF THE TIGER.** Gremlin	●	●		
5	4	↑	**STAINLESS STEEL.** Micro-Gen	●	●		
6	6	↑	**PHANTOMAS II.** Dinamic	●			
7	4	↓	**PYRACURSE.** Hewson	●			
8	15	↑	**WORLD SERIES BASKETBALL.** Imagine	●			
9	15	↑	**BATMAN.** Ocean	●	●		
10	14	↓	**RAMBO.** Ocean	●	●		
11	5	↑	**COBRAS ARC.** Dinamic	●			
12	4	↑	**CAULDRON II.** Palace Soft	●		●	
13	14	↑	**MOVIE.** Imagine	●	●		
14	3	↑	**JACK THE NIPPER.** Gremlin	●	●		●
15	14	↑	**CYBERUN.** ULTIMATE	●			
16	12	↓	**PING PONG.** Imagine	●	●		
17	9	↑	**POLE POSITION.** Atari	●	●		
18	2	↓	**QUAZATRON.** Hewson	●			
19	9	–	**SIR FRED.** Made in Spain	●	●		
20	14	–	**TURBO ESPRIT.** Durell	●	●		

Esta información ha sido elaborada con la colaboración de los centros de Microinformática de El Corte Inglés.

EL COMPACT DISC INTERACTIVO: LA REVOLUCION DE LA INFORMACION

Desde su introducción en 1982, el Compact Disc ha logrado una destacada posición en el mercado y una aceptación en todo el mundo como un soporte de calidad para música y programas de audio.

El desarrollo del sistema CD ha continuado y este soporte ha pasado por una serie de etapas significativas, cada una de las cuales ha ampliado su función original como soporte de música.

Esta evolución ha incluido el aprovechamiento de la capacidad inherente del estándar CD de albergar cantidades limitadas de información, tanto gráfica como en textos. Al mismo tiempo, se desarrolló la primera aplicación del CD como un mero soporte de información para texto o datos, bajo el nombre de CD—ROM (Compact Disc con memoria fija).

En un primer momento se consideró el concepto CD—ROM como un simple soporte para texto y datos, gracias a su tremenda capacidad de memoria de 600 MB. Traducida a texto, esta capacidad representa hasta 300.000 páginas DIN A4.

Se ha recurrido a un sinfín de ejemplos para ilustrar cuánta información podría almacenarse en esta enorme memoria. Un diccionario completo, por ejemplo, ocuparía tan sólo una parte muy pequeña de un disco, mientras que una enciclopedia de varios tomos con texto y gráficos, o un diccionario con miles de ilustraciones en color que incluyese la pronunciación de las palabras, ocuparía un solo disco. La guía telefónica completa de un gran país, comprendidas Páginas Amarillas, códigos postales, etc., podría almacenarse asimismo en un solo disco. Traducido a términos de memoria convencional, 600 MB equivalen a más de 1000 floppy disks normales como los que se emplean hoy día en los ordenadores personales.

Para expresar esta capacidad en otros términos, se ha calculado que la capacidad de un disco equivale a un 1 ó 2 por 100 de la capacidad del cerebro humano. En otras palabras, en un solo disco podría almacenarse algo así como un año de la experiencia propia de una vida humana, información que podría reclamarse instantáneamente. Y ahora se ha dado un nuevo paso en el desarrollo del sistema CD, con la elaboración de la especificación para un nuevo soporte, el CD—I (Compact Disc Interactivo). Muy pronto se establecerá en forma de un estándar concreto. Comparado con el CD—ROM ya existente, el factor clave del CD—I radica en su capacidad de complementar la información básica en forma de texto y datos con imágenes fijas, esquemas, gráficos de gran calidad e imágenes animadas, así como con cualquier tipo de sonido en estéreo, ya se trate de diálogo o de defectos sonoros, todo ello simultáneamente.

Gracias a su capacidad sin precedentes de combinar información de distintos tipos, junto con la plena accesibilidad en cualquier momento de toda la información almacenada, el CD—I representa poco menos que una revolución en el terreno de la publicación electrónica.

El Compact Disc Interactivo es ya una realidad, y sus ilimitadas posibilidades muy pronto van a verse aplicadas en la vida cotidiana: comercios, bibliotecas, escuelas..., cualquier institución o persona que utilice cotidianamente grandes cantidades de datos, podrá aprovecharse de las increíbles ventajas de su uso. El Compact Disc Interactivo ha abierto una nueva generación en el mundo de la información.

RESUMEN DE LAS ESPECIFICACIONES PARA EL CD-INTERACTIVO

1. Formato Físico

Capacidad total de memoria	Aprox. 650 Mbytes
Velocidad de lectura	75 bloques/segundo
Bloque	2352 bytes
Subíndice	8 bytes
Capacidad de memoria para la información del usuario	2048 bytes en Modalidad 2 clase 1, con EDC y ECC 2328 bytes en Modalidad 2 clase 1, sin EDC y ECC

2. Información almacenable

Información en audio y vídeo y datos de ordenador (texto y binarios)
Bloque intercalable

3. Audio

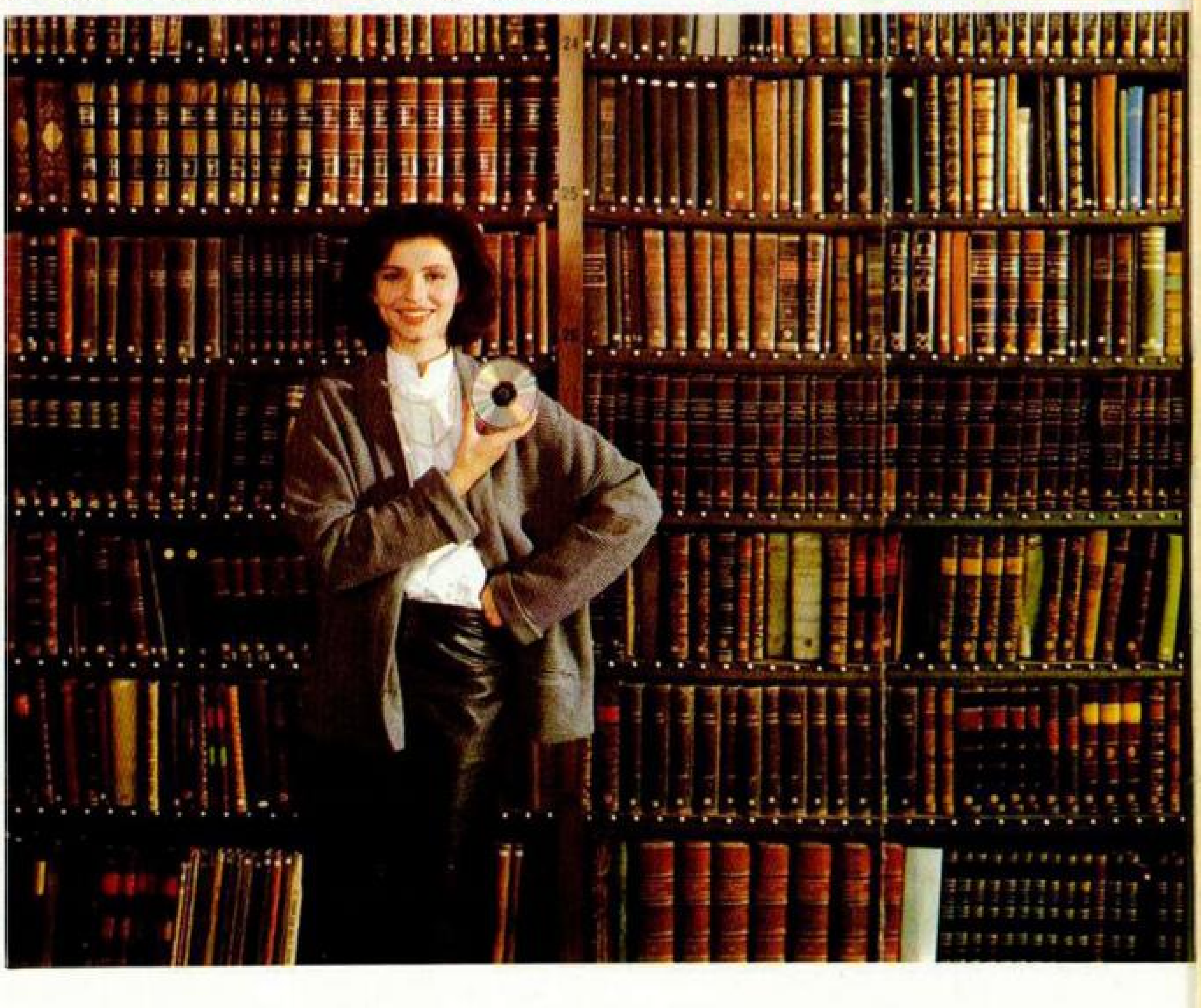
	Estéreo/Mono	Canal
CD-Digital Audio (PCM)	Estéreo	1
Música Alta fidelidad (ADPCM)	Estéreo	2
	Mono	4
Música Fidelidad Media (ADPCM)	Estéreo	4
	Mono	8
Diálogo (ADPCM)	Estéreo	8
	Mono	16

Duración por canal: 70 minutos

4. Video

	Normal	Alta
Resolución de referencia	384×280	768×560

Codificación de pixels:
Imágenes naturales: DYUV (4.2.2)/cuantificación de 5 bytes
Gráficos: (CLUT) 16 colores, 128 colores, 256 colores
(RGB) 256 colores, 32768 colores



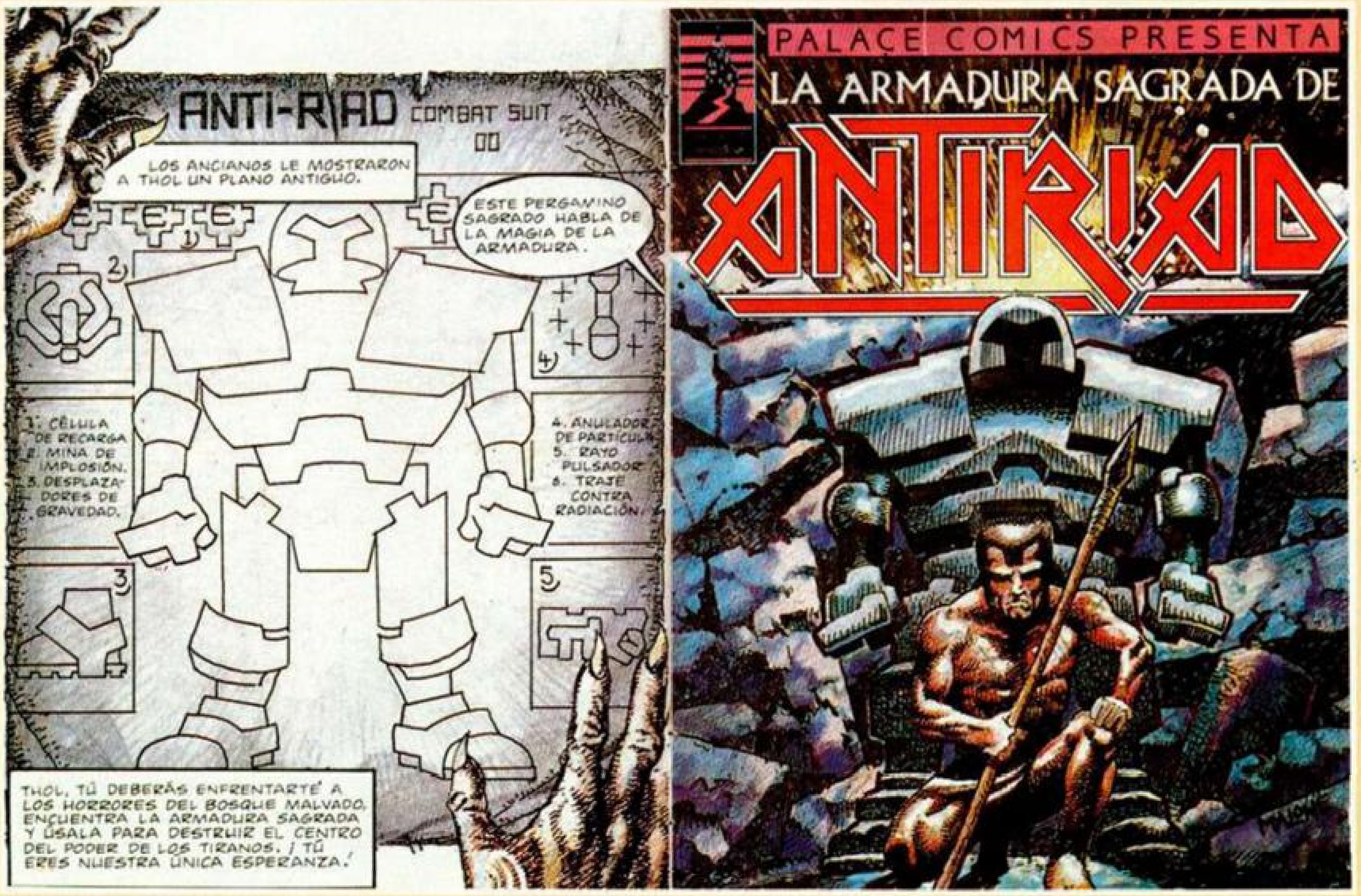
PALACE COMICS PRESENTA



RESUMEN DE LO PUBLICADO

El resurgimiento de una nueva raza tiene lugar tras el holocausto nuclear. La población lleva una vida tranquila y sencilla en medio de una gran superstición hacia un extraño volcán. Un día surge de improviso una fuerza invasora de otro planeta, que ataca y subyuga a los habitantes.





¡ THOL, TÚ ERES LA ÚLTIMA ESPERANZA DE LA HUMANIDAD CONTRA LOS INVASORES!

¡ENCUENTRA LA ARMADURA SAGRADA, DESCUBRE LOS SECRETOS DE SU MAGIA Y ÚSALA PARA DESTRUIR LA FUENTE DEL PODER DEL ENEMIGO!







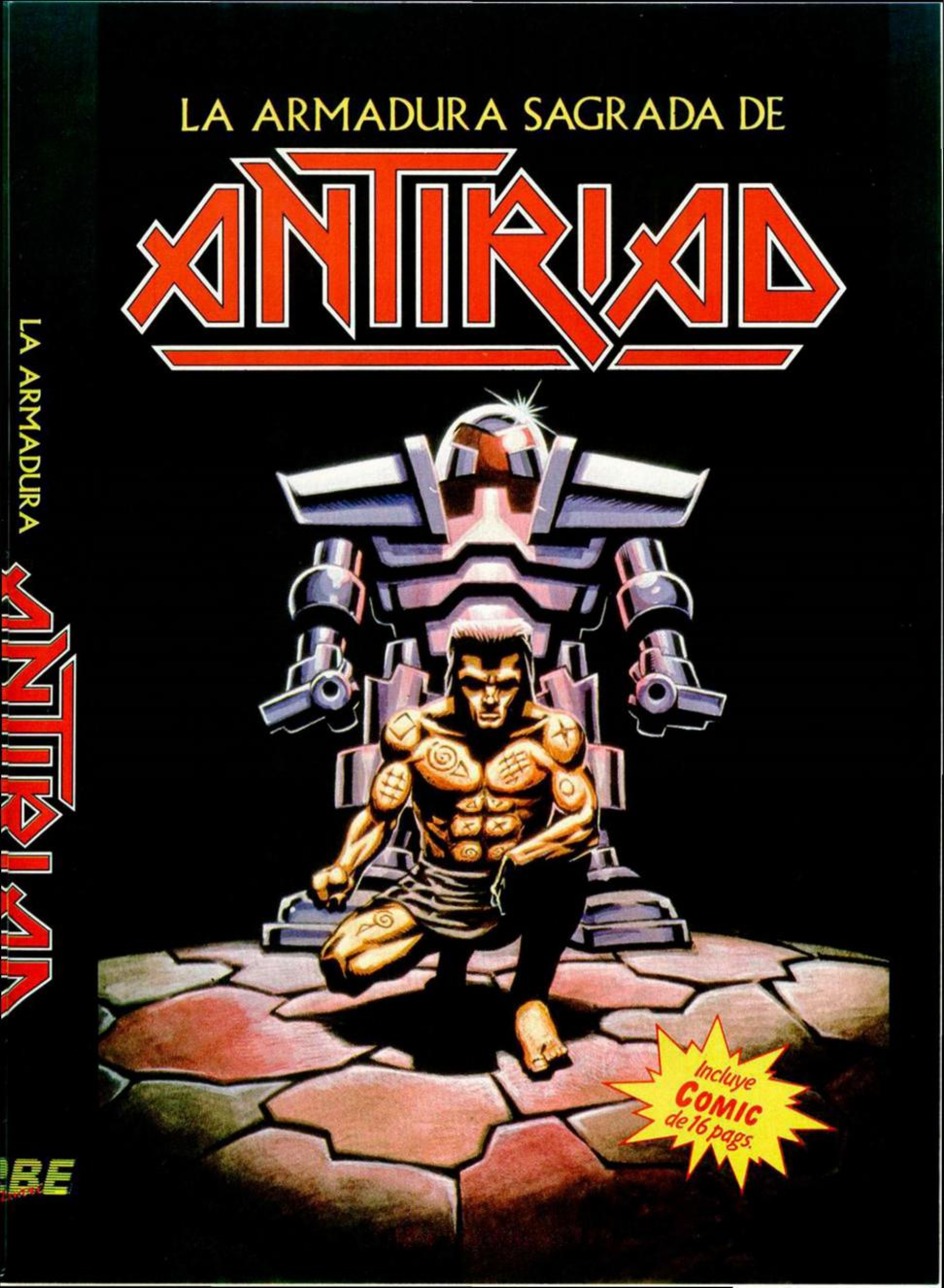
PALACE software

Argumento, diseño, cómic e illustracion de cubierta por Dan Malone.
Programado por Stanley Schembri.
Sonido por Richard Joseph.

ERBE Software

Santa Engracia, 17. 28010 MADRID. (91) 447 34 10

SAGRADA DE

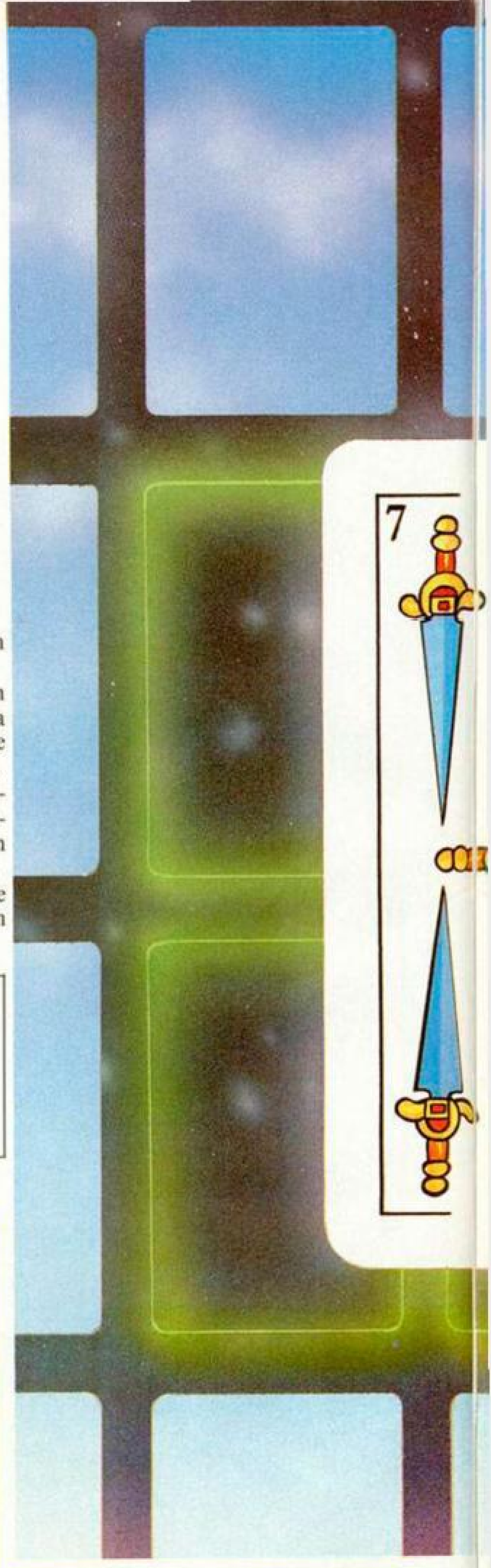

LA ARMADURA SAGRADA DE
ANTIRIAD
LA ARMADURA
ANTIRIAD
Incluye
COMIC
de 16 pags.

SOLITARIO DEL 7

Jorge MORO

SPECTRUM 48 K

DUMP: 40.000
N.º DE BYTES: 5.540



Un juego de cartas siempre puede ayudarnos a pasar el tiempo cuando nos encontremos solos, y más si no necesitamos la colaboración de otros jugadores, siempre difíciles de encontrar. Además, si les ganamos se enfadan y el Spectrum no. Al menos de momento.

Con el juego que ofrecemos a continuación, encontraréis un solitario distinto a los corrientes y cuyas instrucciones de juego pasamos a detallar.

El ordenador, después de barajar, coloca 12 cartas creando en la pantalla un cuadrado de 4×4 y cuatro espacios libres en su interior.

Estos cuatro espacios habrán de ocuparse con los cuatro sietes de la baraja, colocando en orden descendente todas las cartas del mismo palo.

Si entre las cartas que forman el cuadro se encuentra un siete deberemos pasarlo a alguno de los espacios centrales.

Para terminar el juego, deben aparecer en el centro del cuadrado los ases de la baraja y en sus extremos las figuras superiores al siete, es decir, las sotas, los caballos y los reyes.

Para desplazar cada una de las cartas disponemos de tres opciones que son las siguientes:

T. Con esta letra elegimos la opción de *tapete* con la que colocaremos una carta que se encuentra en el cuadro de juego y la posicionaremos en el centro.

P. Con esta opción pasaremos la última carta del pozo a su lugar correspondiente en el *tapete* o, en su caso, en cualquier espacio libre dentro de éste.

M. Al pulsar esta tecla se descubre una nueva carta del mazo y se coloca en el pozo.

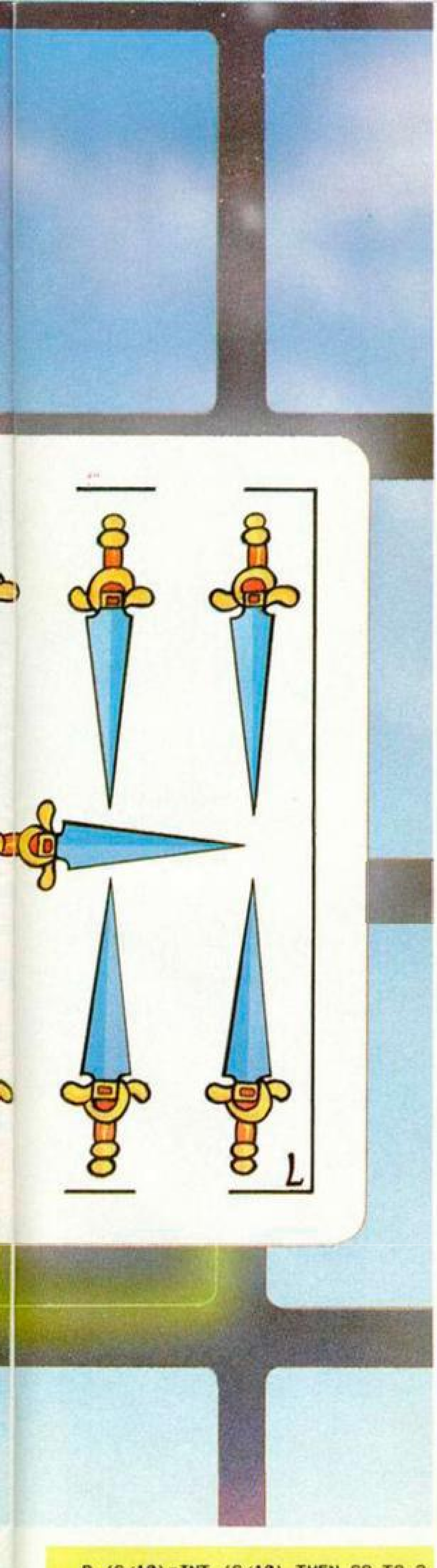
NOTA IMPORTANTE

Todas las letras mayúsculas subrayadas del listado Basic deben teclearse en modo gráfico.

En las líneas que faltan en el listado del cargador de Código Máquina, debe introducirse "0" como datos y "0" como control.

LISTADO 1

```
  10 CLEAR 55000
  20 LOAD ""CODE 55000,5600: CLS
  30 PRINT AT 9,12;"POR FAVOR,";AT 11,7;"AGUARDE UN MOMENTO": DIM b(3,40,1): DIM t(4,4): DIM p(40,1): DIM m(40,1): LET c=40: POKE 23303,3: POKE 23304,5: LET fl=0
  50 FOR n=1 TO 40: FOR m=1 TO 3
  60 READ c1: LET b(m,n,1)=c1
  70 NEXT m: NEXT n
 142 GO SUB 9000: RESTORE : LET PO=0: LET P2=1
 145 BEEP .5,30: CLS : PRINT PAPER 2; INK 7; FLASH 1;AT 10,10;"BARAJANDO"
 146 LET a$="": LET b$="": FOR i=1 TO 40: LET a$=a$+CHR$ i: NEXT i
 150 FOR I=1 TO 40
 160 LET X=INT (RND*40)+1
 165 IF CODE A$(X)<>0 THEN GO TO 180
 170 LET X=X+1: IF X=41 THEN LET X=1
 175 GO TO 165
 180 LET B$=B$+CHR$ X: LET A$(X)=CHR$ 0: NEXT I
 198 LET X1=1
 199 FOR k=1 TO 40
 200 LET m(k,1)=CODE B$(X1): LET X1=X1+1
 205 NEXT K
 208 CLS : PRINT INVERSE 1;AT 10,12;"SUERTE": BEEP .5,30: PAUSE 25
 209 BORDER 4: PAPER 4: BRIGHT 1: CLS
 210 LET Z=0
 220 FOR N=1 TO 4: FOR Q=1 TO 4
 225 IF (n=2 OR n=3) AND (q=2 OR q=3) THEN LET t(n,q)=0: GO TO 240
 230 LET Z=Z+1: LET C=M(Z,1): GO SUB 8000
 235 LET t(N,Q)=C
 240 NEXT Q: NEXT N
 245 FOR N=1 TO 4: PRINT AT 0,3*N-1;N: PRINT AT 5*N-2,0;N: NEXT N
 250 BEEP .1,N: DRAW 255,0: DRAW 0,175: DRAW -255,0: DRAW 0,-175: PLOT 111,0: DRAW 0,175
 261 PRINT #0;AT 0,0;"MAZO(M),POZO(P),TAPETE(T)"
 265 PRINT #0;AT 1,0;"
"
 270 IF INKEY$="t" THEN BEEP .1,17: GO TO 292
 275 IF INKEY$="m" THEN BEEP .1,27: GO TO 1000
 280 IF INKEY$="p" THEN BEEP .1,30: GO TO 2000
 290 GO TO 261
 292 PRINT #0;AT 1,0;"
                         ": PRINT #0;AT 0,0;"
        ": PRINT #0;AT 0,0;"CARTA:"
 293 IF INKEY$="" THEN GO TO 293
 294 LET X$=INKEY$
 295 IF X$="1" THEN BEEP .1,10: GO TO 300
 296 IF X$="2" THEN BEEP .1,20: GO TO 300
 297 IF X$="3" THEN BEEP .1,30: GO TO 300
 298 IF X$="4" THEN BEEP .1,40: GO TO 300
 299 GO TO 293
 300 LET X=VAL X$: PRINT #0;AT 1,0;"(";AT 1,1;X;AT 1,2;","
 305 IF INKEY$<>"" THEN GO TO 305
 310 IF INKEY$="" THEN GO TO 310
 311 LET Y$=INKEY$
 312 IF Y$="1" THEN BEEP .1,10: GO TO 317
 313 IF Y$="2" THEN BEEP .1,20: GO TO 317
 314 IF Y$="3" THEN BEEP .1,30: GO TO 317
 315 IF Y$="4" THEN BEEP .1,40: GO TO 317
 316 GO TO 310
 317 LET Y=VAL Y$: PRINT #0;AT 1,3;Y;AT 1,4;")"
 320 IF (X=2 OR X=3) AND (Y=2 OR Y=3) THEN GO TO 292
 325 LET C=T(X,Y)
 330 IF ((C+2)/10)=INT ((C+2)/10) OR ((C+1)/10)=INT ((C+1)/10) O
```



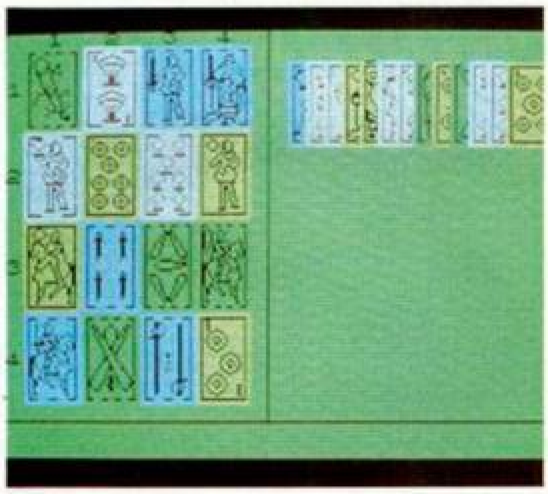
```
R (C/10)=INT (C/10) THEN GO TO 2
60
 350 IF (C+3)/10<>INT (C+3)/10 T
HEN GO TO 800
 400 IF C=7 THEN LET N=X: LET Q=
Y: GO SUB 7500: LET N=2: LET Q=2
: GO SUB 8000: LET T(2,2)=7: LET
 T(X,Y)=0
 500 IF C=17 THEN LET N=X: LET Q
=Y: GO SUB 7500: LET N=2: LET Q=
3: GO SUB 8000: LET T(2,3)=17: L
ET T(X,Y)=0
 600 IF C=27 THEN LET N=X: LET Q
=Y: GO SUB 7500: LET N=3: LET Q=
2: GO SUB 8000: LET T(3,2)=27: L
ET T(X,Y)=0
 700 IF C=37 THEN LET N=X: LET Q
```

```
=Y: GO SUB 7500: LET N=3: LET Q=
3: GO SUB 8000: LET T(3,3)=37: L
ET T(X,Y)=0
 800 FOR I=2 TO 3: FOR J=2 TO 3:
 IF T(I,J)-C=1 THEN GO TO 850
 810 NEXT J: NEXT I
 820 GO TO 260
 850 LET N=X: LET Q=Y: GO SUB 75
00: LET N=I: LET Q=J: GO SUB 800
0
 860 LET T(I,J)=C: LET T(X,Y)=0:
 GO SUB 6000: GO SUB 8500: GO TO
 260
1000 PRINT #0;AT 0,0;"
1001 LET Z=Z+1: IF Z>=41 THEN PR
INT PAPER 7; INK 2;AT 15,15;"NO
HAY MAS CARTAS";AT 17,15;"EN EL
MAZO": BEEP 1.5,-10: GO TO 260
1005 LET C=M(Z,1)
1010 FOR I=1 TO 4: FOR J=1 TO 4
1020 IF (I=2 OR I=3) AND (J=2 OR
 J=3) THEN GO TO 1040
1030 IF T(I,J)=0 THEN GO TO 1060
1040 NEXT J: NEXT I
1050 GO TO 1100
1060 LET N=I: LET Q=J: GO SUB 80
00: LET T(N,Q)=C: GO SUB 8500: G
O TO 260
1140 LET PO=PO+1: LET P1=2: LET
P2=P2+1: IF PO=15 THEN LET P2=2
1145 IF PO>=15 THEN LET P1=8
1150 LET N=P1: LET Q=P2: GO SUB
8030: LET P(PO,1)=C
1160 GO SUB 6000: GO SUB 8500: G
O TO 260
2005 PRINT #0;AT 0,0;"

2010 IF PO<=0 THEN GO TO 260
2020 FOR I=1 TO 4: FOR J=1 TO 4:
 IF (I=2 OR I=3) AND (J=2 OR J=3
) THEN GO TO 2040
2030 IF T(I,J)=0 THEN GO TO 2140
2040 NEXT J: NEXT I
2050 LET C=P(PO,1): IF C=7 THEN
LET I=2: LET J=2: GO TO 2140
2070 IF C=17 THEN LET I=2: LET J
=3: GO TO 2140
2080 IF C=27 THEN LET I=3: LET J
=2: GO TO 2140
2090 IF C=37 THEN LET I=3: LET J
=3: GO TO 2140
2100 FOR I=2 TO 3: FOR J=2 TO 3
2102 LET C=P(PO,1): IF ((C+2)/10
)=INT ((C+2)/10) OR ((C+1)/10)=I
NT ((C+1)/10) OR (C/10)=INT (C/1
0) THEN GO TO 260
2104 IF T(I,J)-P(PO,1)=1 THEN GO
 TO 2140
2105 NEXT J: NEXT I
2110 GO TO 260
2140 LET n=8: IF po<=14 THEN LET
 n=2
2142 IF p2=1 THEN LET p2=15
2145 LET q=p2: LET C=P(PO,1): GO
 SUB 7600
2150 IF PO=1 OR PO=15 THEN GO TO
 2500
2160 LET N=8: IF PO-1<=14 THEN L
ET N=2
2170 LET Q=P2-1
2180 LET C=P(PO-1,1): GO SUB 803
0
2500 LET N=I: LET Q=J: LET C=P(P
O,1): GO SUB 8000: LET T(I,J)=C:
 LET PO=PO-1: LET P2=P2-1
2505 GO SUB 6000: GO SUB 8500: G
O TO 260
6000 LET TOT=0
6001 FOR I=2 TO 3: FOR J=2 TO 3:
 LET FI=T(I,J): IF ((FI+9)/10)<>
INT ((FI+9)/10) THEN RETURN
6007 NEXT J: NEXT I
6008 FOR I=1 TO 4: FOR J=1 TO 4
6010 LET U=T(I,J): LET TOT=TOT+U
: NEXT J: NEXT I
6020 IF TOT=352 THEN GO TO 7000
6030 RETURN
7001 PRINT AT 15,15;"
      ";AT 17,15;"
 "
7002 PRINT PAPER 2; INK 7; FLASH
 1;AT 5,17;"ENHORABUENA";AT 10,1
5;"LO HA CONSEGUIDO": RESTORE 81
50
7003 LET CONT=0: LET c=.25: LET
d=.5: LET e=.75: LET g=1.5: LET
h=1.75
7004 LET i=0: LET j=1: LET k=2:
LET l=3: LET m=4: LET n=5: LET o
=6: LET p=7: LET q=8: LET r=9: L
ET s=10: LET t=11: LET u=12: LET
 v=13: LET w=14: LET x=15: LET y
=16: LET z=17: LET aa=19: LET bb
=21
7010 READ A,B: LET CONT=CONT+1:
IF CONT=693 THEN LET CONT=0: RES
TORE 8150
7020 IF a=i THEN GO SUB 7035
7030 BEEP a/1.9,b
7035 IF INKEY$<>"" THEN GO TO 14
2
7040 GO TO 7010
7050 IF b=i THEN CLS : PAUSE 500
0: GO TO 142
7060 PAUSE b: RETURN
7510 POKE 23305,96: POKE 23306,2
16: POKE 23307,236: POKE 23302,1
: POKE 23300,3*Q-2: POKE 23301,5
*N-4: RANDOMIZE USR 60400
7520 RETURN
7600 POKE 23305,96: POKE 23306,2
16: POKE 23307,236: POKE 23302,1
: POKE 23300,Q+13: POKE 23301,N:
 RANDOMIZE USR 60400: RETURN
8010 POKE 23305,B(3,C,1): POKE 2
3306,B(1,C,1): POKE 23307,B(2,C,
1): POKE 23302,1: POKE 23300,3*Q
-2: POKE 23301,5*N-4: RANDOMIZE
USR 60400
8020 RETURN
8030 POKE 23305,B(3,C,1): POKE 2
3306,B(1,C,1): POKE 23307,B(2,C,
1): POKE 23302,1: POKE 23300,Q+1
3: POKE 23301,N: RANDOMIZE USR 6
0400: RETURN
```

```
8100 DATA 216,214,48,80,215,48,2
00,215,48,64,216,48,184,216,48,4
8,217,48,168,217,48,248,227,48,1
12,228,48,232,228,48
8110 DATA 32,218,58,152,218,58,1
6,219,58,136,219,58,0,220,58,120
,220,58,240,220,58,96,229,58,216
,229,58,80,230,58
8120 DATA 104,221,40,224,221,40,
88,222,40,208,222,40,72,223,40,1
92,223,40,56,224,40,200,230,40,6
4,231,40,184,231,40
8130 DATA 176,224,32,40,225,32,1
60,225,32,24,226,32,144,226,32,8
,227,32,128,227,32,48,232,32,168
,232,32,32,233,32
8150 DATA c,k,c,l,c,m,d,u,c,m,d,
u,c,m,g,u,c,u,c,w,c,x,c,y,c,u,c,
w,d,y,c,t,d,w,g,u
8151 DATA c,k,c,l,c,m,d,u,c,m,d,
u,c,m,g,u,c,r,c,p,c,o,c,r,c,u,d,
y,c,w,c,u,c,r,g,w
8152 DATA c,k,c,l,c,m,d,u,c,m,d,
u,c,m,g,u,c,u,c,w,c,x,c,y,c,u,c,
w,d,y,c,t,d,w,g,u
8153 DATA c,u,c,w
8154 DATA c,y,c,u,c,w,d,y,c,u,c,
w,c,u
8155 DATA c,y,c,u,c,w,d,y,c,u,c,
w,c,u
8156 DATA c,y,c,u,c,w,d,y,c,t,d,
w,g,u
8157 DATA c,k,c,l,c,m,d,u,c,m,d,
u,c,m,g,u,c,u,c,w,c,x,c,y,c,u,c,
w,d,y,c,t,d,w,g,u
8158 DATA c,k,c,l,c,m,d,u,c,m,d,
u,c,m,g,u,c,r,c,p,c,o,c,r,c,u,d,
y,c,w,c,u,c,r,g,w
8159 DATA c,k,c,l,c,m,d,u,c,m,d,
u,c,m,g,u,c,u,c,w,c,x,c,y,c,u,c,
w,d,y,c,t,d,w,g,u,c,u,c,w
8160 DATA c,y,c,u,c,w,d,y,c,u,c,
w,c,u
8161 DATA c,y,c,u,c,w,d,y,c,u,c,
w,c,u
8162 DATA c,y,c,u,c,w,d,y,c,t,d,
w,j,u
8220 DATA i,20,c,m,c,n,c,o
8221 DATA d,p,c,r,d,p,c,m,c,n,c,
o
8222 DATA d,p,c,r,d,p,c,y,c,u,c,
p,c,r,c,t,c,u,c,w,c,y,c,w,c,u,c,
w,j,p
8223 DATA i,p,c,p,c,m,c,n
8224 DATA d,p,c,r,d,p,c,m,c,n,c,
o
8225 DATA d,p,c,r,c,p,c,r,c,s,c,
t,d,t,d,t,c,r,c,o,c,k,j,p,i,p
8226 DATA i,p,c,m,c,n,c,o
8227 DATA d,p,c,r,d,p,c,m,c,n,c,
o
8228 DATA d,p,c,r,d,p,c,y,c,u,c,
p,c,r,c,t,c,u,c,w,c,y,c,w,c,u,c,
w,1.25,u
8230 DATA c,p,c,o,c,p,d,u,c,r,d,
u,c,r,c,u,c,r,c,p,c,u,c,y,d,aa,c
,y,c,u,c,p,d,r,d,u,c,y,e,w,j,u,i
,n
8231 DATA c,k,c,l,c,m,d,u,c,m,d,
u,c,m,g,u,c,u,c,w,c,x,c,y,c,u,c,
w,d,y,c,t,d,w,g,u
8232 DATA c,k,c,l,c,m,d,u,c,m,d,
u,c,m,g,u,c,r,c,p,c,o,c,r,c,u,d,
y,c,w,c,u,c,r,g,w
8233 DATA c,k,c,l,c,m,d,u,c,m,d,
u,c,m,g,u,c,u,c,w,c,x,c,y,c,u,c,
w,d,y,c,t,d,w,g,u
8234 DATA c,u,c,w
8235 DATA c,y,c,u,c,w,d,y,c,u,c,
w,c,u
8236 DATA c,y,c,u,c,w,d,y,c,u,c,
w,c,u
8237 DATA c,y,c,u,c,w,d,y,c,t,d,
w,g,u
8250 DATA c,r,c,q,j,r,d,u,k,w,c,
n,c,m,j,n,d,r,h,s,c,p
8251 DATA d,k,c,p,d,k,c,p,d,k,j,
i,j,n,c,m,c,q,c,t,d,y,c,w,c,t,c,
u,j,r,j,s
8252 DATA c,r,c,q,j,r,d,u,k,w,c,
n,c,m,j,n,d,r,h,s,c,p
8253 DATA d,k,c,p,d,k,c,p,d,k,j,
i,e,n,c,n,c,r,d,u,d,p,c,k,c,m
8254 DATA d,n,c,-j,c,i,c,k,c,m,c
,n,c,p
8255 DATA c,r,c,q,j,r,d,u,k,w,c,
n,c,m,j,n,d,r,h,s,c,p
8256 DATA d,k,c,p,d,k,c,p,d,k,j,
i,j,n,c,m,c,q,c,t,d,y,c,w,c,t,c,
u,j,r,j,s
8257 DATA c,r,c,q,j,r,d,u,k,w,c,
n,c,m,j,n,d,r,h,s,c,p
8258 DATA d,k,c,p,d,k,c,p,d,k,j,
i,e,n,c,n,c,r,d,u,d,p,c,k,c,m
8260 DATA d,n,i,s,d,w,i,s
8270 DATA d,u,c,r,d,u,c,r,c,u,c,
r,c,p,c,u,c,y,d,aa,c,y,c,u,c,p,d
,r,d,u,c,y,e,w,j,u,d,u,i,s
8271 DATA d,n,c,m,d,n,c,m,d,n,i,
p,c,u,c,y,c,u,c,w,c,y,c,w,c,u
8272 DATA d,w,c,v,d,w,c,v,d,w,i,
p,c,z,c,bb,c,z,c,aa,c,bb,c,aa,c,
z,c,u,c,u,j,u,d,r,d,p,c,p,d,p,d,
p
8273 DATA d,n,c,m,d,n,c,m,d,n,i,
p,c,u,c,y,c,u,c,w,c,y,c,w,c,u
8274 DATA c,r,c,q,c,r,d,aa,d,z,c
,u,c,y,c,x,c,y,d,bb,c,24,c,aa,c,
y,d,u,d,u,c,y,e,w
8275 DATA d,u,c,p,c,p,d,p,d,p
8276 DATA d,u,c,r,d,u,c,r,c,u,c,
r,c,p,c,u,c,y,d,aa,c,y,c,u,c,p,d
,r,d,u,c,y,e,w,j,u,d,u,i,s
8277 DATA d,n,c,m,d,n,c,m,d,n,i,
p,c,u,c,y,c,u,c,w,c,y,c,w,c,u
8278 DATA d,w,c,v,d,w,c,v,d,w,i,
p,c,z,c,bb,c,z,c,aa,c,bb,c,aa,c,
z,c,u,c,u,j,u,d,r,d,p,c,p,d,p,d,
p
8279 DATA d,n,c,m,d,n,c,m,d,n,i,
p,c,u,c,y,c,u,c,w,c,y,c,w,c,u
8280 DATA c,r,c,q,c,r,d,aa,d,z,c
,u,c,y,c,x,c,y,d,bb,c,24,c,aa,c,
y,d,u,d,u,c,y,e,w
```

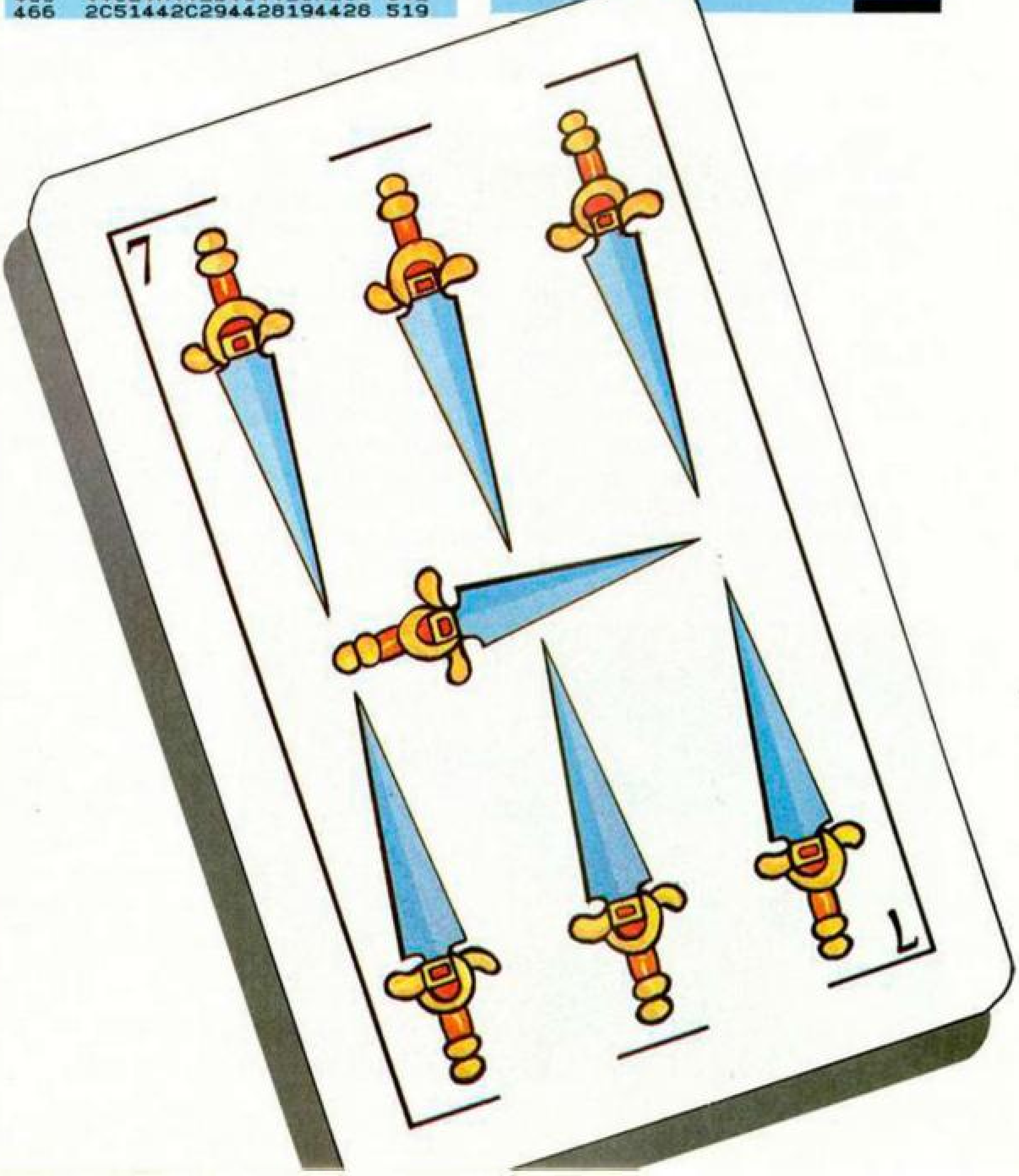
```
8290 DATA j,u,d,u,i,u
8291 DATA c,k,c,l,c,m,d,u,c,m,d,
u,c,m,g,u,c,u,c,w,c,x,c,y,c,u,c,
w,d,y,c,l,d,w,g,u
8292 DATA c,k,c,l,c,m,d,u,c,m,d,
u,c,m,g,u,c,r,c,p,c,o,c,r,c,u,d,
y,c,w,c,u,c,r,g,w
8293 DATA c,k,c,l,c,m,d,u,c,m,d,
u,c,m,g,u,c,u,c,w,c,x,c,y,c,u,c,
w,d,y,c,l,d,w,g,u
8294 DATA c,u,c,w
8295 DATA c,y,c,u,c,w,d,y,c,u,c,
w,c,u
8296 DATA c,y,c,u,c,w,d,y,c,u,c,
w,c,u
8297 DATA c,y,c,u,c,w,d,y,c,l,d,
w,k,u,i,i
8500 IF Z<40 THEN RETURN
8501 IF PO=0 THEN GO TO 8530
8502 IF ((P(PO,1)+3)/10)=INT ((P
(PO,1)+3)/10) THEN RETURN
8503 FOR i=1 TO 4: FOR j=1 TO 4:
 IF (i=2 OR i=3) AND (j=2 OR j=3
) THEN GO TO 8506
8504 IF ((t(i,j)+3)/10)=INT ((t(
i,j)+3)/10) THEN RETURN
8505 IF t(i,j)=0 THEN RETURN
8506 NEXT j: NEXT i
8507 FOR I=2 TO 3: FOR J=2 TO 3
8508 IF T(I,J)-P(PO,1)=1 THEN RE
TURN
8510 NEXT J: NEXT I
8530 FOR M=2 TO 3: FOR N=2 TO 3
8540 FOR I=1 TO 4: FOR J=1 TO 4
8545 IF ((t(m,n)+9)/10)=INT ((t(
m,n)+9)/10) THEN GO TO 8560
8550 IF T(M,N)-T(I,J)=1 THEN RET
URN
8560 NEXT J: NEXT I: NEXT N: NEX
T M
8570 PRINT AT 15,15;"LO SIENTO,
NO"; PAPER 4;"     ";AT 17,15;"PU
EDES SEGUIR.";AT 19,15;"PRUEBA D
E NUEVO!": BEEP 1,-10
8580 IF INKEY$<>"" THEN CLS : GO
 TO 142
8581 GO TO 8580
9010 CLS : RETURN
```



LISTADO 2

Línea	Datos	Control
1	3E00000000003FFFFC20	664
2	0004207E042781E4283C	662
3	1425C3A4263C6430C30C	869
4	2D3EB42E5974244D2424	723
5	A6A4288294284294284 9	1015
6	14287214242124244D24	448
7	2280C421FF8420C30431	1106
8	3C8C3A425C34A52C3318	752
9	CC3C003C2FFFF4297854	1118
10	297AD42B4A542FFFF420	1154
11	18042E66742E18742000	510
12	043FFFFC000000000000	574
13	00000000000003FFFFC20	602
14	00042000042018042066	234
15	04208104211884212484	559
16	222A442262442114842 1	562
17	14842081042066042018	511
18	04200004200004200004	112
19	20000420000420000420	140
20	18042066042081042118	388
21	84212484222A44226244	677
22	21148421148420810420	567
23	66042018042000042000	234
24	043FFFFC000000000000	574
25	0000000000003FFFFC2C	614
26	00042400042C00042400	128
27	042CC004233004240804	379
28	24C804292404 28A40424	565
29	C80424080423300420C0	563
30	04200304200CC4201024	367
31	20132420249420229420	549
32	1324201024200CC420C3	606
33	04233004240804 24C804	379
34	29240028A43424C82424	641
35	0834233024 20C0342000	487
36	043FFFFC000000000000	574
37	00000000000003FFFFC20	602
38	00042800042C00042400	132
39	042000042381C4244224	538
40	2924942AA55429249424	777
41	42242381C42000042000	530
42	0420A80420E80420A804	680
43	20000422938421110420	435
44	00042000042381C42442	502
45	242924942AA554292494	777
46	2442242381C420000420	566
47	00242000342000142000	204
48	043FFFFC000000000000	574
49	00000000000003FFFFC20	602
50	00042C00042400042C00	136
51	042381C4244224292494	727
52	2AA5542924942442242 3	689
53	81C42000042018042066	555
54	04208104209904212484	559
55	21148420990420810420	571
56	66042018042000042381	366
57	C4244224292494 2AA554	850
58	2924942442242381C420	755
59	00342000242000342000	236
60	043FFFFC000000000000	574
61	00000000000003FFFFC20	602
62	00042800042381C42442	510
63	242924942AA554292494	777
64	2442242381C420000420	566
65	00042000042000042381	240
66	C42442242924942AA554	850
67	2924942442242381C420	755
68	00042000042000002000	104
69	042000042381C4244224	538
70	2924942AA55429249424	777
71	42242381C42000142000	546
72	043FFFFC000000000000	574
73	00000000000003FFFFC20	602
74	00042800042381C42442	510
75	242924942AA554292494	777
76	2442242381C420380420	622
77	44042092042 0AA042092	638
78	04204404203804200004	236
79	2381C424422429249 42A	765
80	A5542924942442242381	776
81	C42000042381C4244224	730
82	2924942AA55429249424	777
83	42242381C42000142000	546
84	043FFFFC000000000000	574
85	00000000000003F00FC20	347
86	00042818042824042818	216
87	04281804202404205204	262
88	20890421008 4223C4425	537
89	C3A42E00742800142524	654
90	A424A52422A544221844	794
91	21668420810420660420	602
92	24042018042018042024	228
93	04205A04202404201804	262
94	2018042 03C04207E0420	350
95	AA0421FF8427FFE42000	1148
96	043F00FC000000000000	319
97	00000000000003F00FC20	347
98	00042C18042466042D81	392
99	84220044227E44218184	756
100	21008420A504205A0420	524
101	4204203C04201804203C	318
102	04205A0420FF04200004	457
103	20000420180420660421	267
104	8184220044227E442181	753
105	8421008420A504205A04	624
106	204204203C0420180420	290
107	3C34205A2420FF342000	641
108	043F00FC000000000000	319
109	00000000000003F00FC20	347
110	000428800439C0042220	495
111	0424100429C804263004	395
112	22200421400420800421	368
113	C00422A10427F3842004	845
114	442008242013942 00C64	487
115	20044420028420010420	339
116	0384208544 21CFE42220	902
117	0424100429C804263004	395
118	22200421403420802421	448
119	C03422A02427F0342000	837
120	043F00FC000000000000	319
121	00000000000003F00FC20	347
122	00042100842281442442	502
123	242924942 6C364228144	825
124	21008423 81C427C3E420	1019
125	00042000042000042000	108
126	04276D94216D94215554	792
127	25555425553427553420	588
128	00042000042000042000	108
129	04210084228144244224	538
130	2924942 6C36422814421	822
131	0084228144 27C3E42000	857
132	043F00FC000000000000	319
133	00000000000003F00FC20	347
134	00042100842281442442	502
135	242924942 6C364228144	825
136	2100842281442 7C3E420	890
137	00042010042038042044	248
138	04208204213904 20C604	498
139	20440420280420100420	264
140	3804205404 20FE042000	502
141	04210084228144244224	538
142	2924942 6C36422814421	822
143	0084228144 27C3E42000	857
144	043F00FC000000000000	319
145	00000000000003F00FC20	347
146	00042100842281442442	502
147	24282414244224228144	501
148	21008423 81C427C3E420	1019
149	00042000042000042100	109
150	84228144244224282414	597
151	24422422814421008422	568
152	814427C3E42000042000	727
153	04210084228144244224	538
154	28241424422422814421	498
155	0084228144 27C3E42000	857
156	043F00FC000000000000	319
157	00000000000003F00FC21	348
158	00842281442442242824	577
159	14244224228144210084	554
160	22814427D3E420280420	817
161	44042082042044042028	414
162	04201004212884 22FD44	616
163	24422428241424422422	406
164	8144210084228144 27C3	827
165	E4210084228144244224	762
166	28241424422422814421	498
167	0084228144 27C3E42000	857
168	043F00FC000000000000	319
169	00000000000003C3C3C20	212
170	00842F01642CC2742A20	708
171	643120A4309EA4287EA4	1045
172	24F8A425E92427C92427	1069
173	4A64204A94204A94204C	790
174	94205C4C212C4C22A454	783
175	22645422249421652421	639
176	66242296742 29A542282	922
177	04225204221204240904	229
178	2409042809042804843C	338
179	04843402842002842001	521
180	843C3C3C000000000000	312
181	00000000000003C3C3C20	212
182	00042380442340C42320	597
183	C42520C42520C42340C4	1021
184	2F80C42800C42300C423	876
185	00C42300C42300C42300	693
186	C42310C42318C42310C4	945
187	2300C42318C42300C423	752
188	10C42308C42310C42300	733
189	C42300C42300D42301F4	954
190	2302C42304A42304A423	674
191	04C42302C42201C42000	696
192	043C3C3C000000000000	184
193	00000000000003C3C3C20	212
194	00042708E42698D42658	807
195	CC2A594C2A594C2698D4	1020
196	3F1BE4361AC42618C426	890
197	18C42618C42618C42618	798
198	C42618C42618C42618C4	970
199	2618C42618C42618C426	812
200	18C42618C42618C42618	798
201	C42618C4261AC4263EC4	1010
202	2658C42694C42694C426	1124
203	98C42658C42438842000	926
204	043C3C3C000000000000	184
205	00000000000003C3C3C20	212
206	00042000042300C42280	433
207	A42300C42781E42300C4	1022
208	2300C42300C42300C423	728
209	00C42300C42300C42200	692
210	84220084200004200004	370
211	20000420000423 00C422	337
212	80A42300C42781E42300	954
213	C42300C42300C42300C4	889
214	2300C42300C42300C422	727
215	00842200842000042000	366
216	043C3C3C000000000000	184
217	00000000000003C3C3C20	212
218	00042300C42280A42300	596
219	C42781E42300C42300C4	1054
220	2300C42300C42300C423	728
221	00C42300C42200842200	627
222	842000042002 04 27FFC4	696
223	21FF442002842000042 3	593
224	00C42280A42300C42781	921
225	E42300C42300C42300C4	921
226	2300C42300C42300C423	728
227	00C42200842200842000	560
228	043C3C3C000000000000	184
229	00000000000003C3C3C20	212
230	00042000042618642514	259
231	542618642F3CF4261864	759
232	26186426186426186426	524
233	18642618642618642410	500
234	44241044200004200004	260
235	20000420000426186425	271
236	14542618642F3CF42618	679
237	64261864261864261864	586
238	26186426186426186424	522
239	10442410442000042000	272
240	043C3C3C000000000000	184
241	00000000000003C3C3C20	212
242	00042618642514542618	369
243	642F3CF4261864261864	775
244	26186426186426186426	524
245	18642618642410442410	458
246	442000042002 04 27FFC4	632
247	21FF442002842000042 6	596
248	186425145426186 42F3C	534
249	F4261864261864261864	730
250	26186426186426186426	524
251	18642410442410442000	396
252	043C3C3C000000000000	184
253	00000000000038E71C20	347
254	00042300042480042840	315
255	04282004282604286604	308
256	24A80425590426960423	565
257	11042210042210042108	170
258	04210804212804 20A404	326
259	20CCA424D444236AA424	1057
260	33042422042811042811	247
261	44240924204944202484	522
262	201D84202344204124 20	493
263	01E42000642000042000	429
264	0438E71C000000000000	319
265	00000000000038E71C20	347
266	00042000042181442661	405
267	442421442C3144281144	491
268	28124428114428334428	450
269	D2442934A42A55242C96	892
270	24252424262424242424	363
271	24242424246424 24A424	552
272	693424AA54252C94224B	785
273	142ACC142748142A4814	551
274	22881422 8C3422842422	652
275	86642281842100042000	598
276	0438E71C000000000000	319
277	00000000000038E71C20	347
278	00042000042C18342A24	238
279	54264264 2542A42542A4	822
280	22C3442242442124 8421	699
281	248420990420990 4204A	652
282	04204604202404206204	316
283	20620420910421108421	529
284	18842218C42424242424	590

```
285   24285A14285A14305A0C   486
286   30990C31321829189426   587
287   18642000042000042000   228
288   0438E71C000000000000   319
289   0000000000000038E71C20 347
290   00042700E42900942481   625
291   24248124224244224244   573
292   214284214284200C30420  725
293   A5042066042042042000   441
294   04200004200004200004   112
295   20000420000420000420   140
296   000420420420660420A5   441
297   0420C304214284214284   697
298   22424422424424812424   573
299   8124290094270 0E42000  653
300   0438E71C000000000000   319
301   0000000000000038E71C20 347
302   000420240420 3C04205A  294
303   0420A50420A504214284   637
304   22424422424424812428   577
305   811428811429009 42E00  573
306   7420FC042103F4220034   770
307   2103C420FC042000042E   602
308   00742900942881142881   663
309   14248124224244224244   557
310   21428420A50420A50420   665
311   5A04203C042024042000   294
312   0438E71C000000000000   319
313   0000000000000038E71C20 347
314   0004220824251454 28A2  425
315   8C28A28C28A28C28A28C   1166
316   28A28C28A28C25145425   862
317   14542514542514542000   418
318   04200004200004200004   112
319   2000042000042000042A   150
320   28A42A28A42A28A42A28   778
321   A4314514314514314514   578
322   31451431451431451431   463
323   45142A28A42410442000   487
324   0438E71C000000000000   319
325   0000000000000038E71C20 347
326   00042208242514542 8A2  425
327   8C28A28C28A28C28A28C   1166
328   28A28C28A28C25145425   862
329   14542514542514542000   418
330   0421F8042207E4240064   694
331   22078421F80420000420   526
332   00042A2AA42A28A42A28   578
333   A42A28A4314514314514   686
334   31451431451431451431   463
335   45142A28A42410442000   487
336   0438E71C000000000000   319
337   00000000000003FFFFC20  602
338   0004230C042492042859   370
339   04285104249904231304   380
340   202E84205084208944 21  724
341   26A42261242 4A2242326  676
342   44204984204884 20A844  809
343   20904420804420408420   732
344   7F84202104202D04202C   485
345   84202A84202A84202A84   750
346   202A44202A4420294420   457
347   29442029E42078042000   598
348   043FFFFC000000000000   574
349   00000000000003FFFFC23  605
350   00042486042849042852   417
351   04248A04231204 2E2F04  336
352   29A28428448428458432   866
353   5984343284344C442A82   823
354   44231244223234244A1C   463
355   268A1C250A1C25052C28   405
356   052C28054C2C028C227E   516
357   8C22524C22A2D4234534   896
358   228A3422943422981423   699
359   10142200142600342000   212
360   043FFFFC000000000000   574
361   00000000000003FFFFC20  602
362   00042305042487042848   335
363   84284D04248984233A44   719
364   20642420501420881421   521
365   09142214542722142FBE   497
366   8C2DF10C28A98C2D496C   1013
367   27667C2250CC224D4C22   804
368   5354225254225254224B   676
369   542255542260D4227FD4   1002
370   2140542140642FBFC428   852
371   604430DF842B09042419   684
372   843FFFFC000000000000   702
373   00000000000003F00FC27  354
374   0004288C043052042899   515
375   04251104221904251304   185
376   2FAE8425508424894422   877
377   26A42161242 0A2242026  668
378   44204984204A8420AD44   816
379   20914420894420468420   748
380   7F84202104202D04202C   485
381   84202A84202A84202A84   750
382   202A44202A4420294420   457
383   29442029E42078042000   598
384   043F00FC000000000000   319
385   00000000000003F00FC22  349
386   00042506042889042512   287
387   04220A042F92042E2F04   346
388   29A284285484286D8432   922
389   5984343284345C442AAA   879
390   44231644223A34244E1C   479
391   268A1C254E1C25452C28   537
392   272C28254C2C428C227E   646
393   8C22524C22A2D4234534   896
394   228A3422943422981423   699
395   10142200142600342000   212
396   043F00FC000000000000   319
397   00000000000003F00FC20  347
398   00042205042507042888   271
399   84250D04220984273A44   526
400   20642420501420885421   585
401   09142214542722142DBE   495
402   4C29F10C2DA98C2D496C   950
403   27667C2250CC224D4C22   804
404   5354225254225254224B   676
405   542255542260D4227FD4   1002
406   2140542140642FBFC428   852
407   604430DF842B09042419   684
408   843F00FC000000000000   447
409   00000000000003C3C3C20  212
410   0004240C042412042C19   183
411   042C11042C19042C1304   209
412   2C2E842C50842C89442D   772
413   26A42E61242CA2242D26   706
414   442E49843E4A842CAD44   872
415   3491442C894420468420   780
416   7F84202104202D04202C   485
417   84202A84202A84202A84   750
418   202A44202A4420294420   457
419   29442029E42078042000   598
420   043C3C3C000000000000   184
421   00000000000003C3C3C20  212
422   00042460042490042448   432
423   04245004244824 24F454  632
424   254594252A14258614 2F  639
425   9A4C254C2C263A2C22D5   774
426   542368C42C4C44387224   813
427   3851643852A434A2A434   969
428   E41432A414314234317E   824
429   44324A442B45442CA2C4   842
430   2C51442C294428194428   519
431   00C42800442C00642000   480
432   043C3C3C000000000000   184
433   00000000000003C3C3C20  212
434   00042405042607042608   144
435   84260D04260984263A44   530
436   26642426511426885427   610
437   09142614542722142FBE   501
438   4C2CF10C3353182D496C   757
439   27667C2250CC224D4C22   804
440   5354225254225254224B   676
441   542255542260D4227FD4   1002
442   2140542140642FBFC428   852
443   604430DF842B09042419   684
444   843C3C3C000000000000   312
445   0000000000000038E71C20 347
446   0004260C042912042919   187
447   04291104291904291304   200
448   2A2E842A50842A894427   760
449   26A42661242 4A2242526  682
450   44264984204A8420AD44   822
451   20914420894420468420   748
452   7F84202104202D04202C   485
453   84202A84202A84202A84   750
454   202A44202A4420294420   457
455   29442029E42078042000   598
456   0438E71C000000000000   319
457   0000000000000038E71C20 347
458   00042C60043290043248   468
459   04325004324824 32F454  674
460   354594352A1435861429   681
461   9A4C294C2C223A2C22D5   774
462   542368C42C4C44387224   813
463   3851643852A434A2A434   969
464   E41432A414314234317E   824
465   44324A442B45442CA2C4   842
466   2C51442C294428194428   519
467   00C42800442C00642000   480
468   0438E71C000000000000   319
469   0000000000000038E71C20 347
470   00042605042907042908   152
471   84290D04290984293A44   539
472   2A64242A51142A88542D   628
473   09142E14542F22142FBE   517
474   4C2CF10C29A98C2D496C   949
475   27667C2250CC224D4C22   804
476   5354225254225254224B   676
477   542255542260D4227FD4   1002
478   2140542140642FBFC428   852
479   604430DF842B09042419   684
480   8438E71C000000000000   447
483   0000000000000000007F00 127
484   007F00007F00001C0000   282
485   1C00001C00001C00001C   112
486   00001C00001C00001C00   84
487   001C00001C00001C0000   84
488   1C00001C00001C00007F   211
489   00007F00007F00000000   254
495   0000000000000007FFE0   486
496   07FFE007FFE001C38001   1297
497   C38001C38001C38001C3   1167
498   8001C38001C38001C380   1100
499   01C38001C38001C38001   973
500   C38001C38001C38007FF   1233
501   E007FFE007FFE0000000   1196
507   00000000000000 07FFFFC 634
508   7FFFFC7FFFFC1C38701C   1492
509   38701C38701C38701C38   644
510   701C38701C38701C3870   700
511   1C38701C38701C38701C   616
512   38701C38701C38707FFF   942
513   FC7FFFFC7FFFFC000000   1520
519   000000000000003FFFFC   570
520   3FFFFC3FFFFC0E38380E   1280
521   38380E38380E1C700E1C   434
522   700E1C700E0EE00E0EE0   770
523   0E0EE00E07C00E07C00E   692
524   07C00E03800E03803FFF   807
525   FC3FFFFC3FFFFC000000   1392
541   F3DD21045BED4B045B78   1119
542   0F0F0FE6E06F78E618F6   1230
543   40670600095D7C0F0F0F   444
544   E603F65857D5ED5B0A5B   1296
545   3A085B080E08E5DD4603   710
546   DDCB024620031100 3D1A  635
547   77132310FAE1240D20E8   977
548   7DE6E0FEE0280801E007   1337
549   A7ED4218097CFE573004   1020
550   7DE61F6F083D20C9E111   1041
551   2000DDCB024620053A12   641
552   5B18077E32125B3A095B   565
553   DD4E04DD4603E5772310   996
554   FCE1190D20F3FBC90000   1242
```

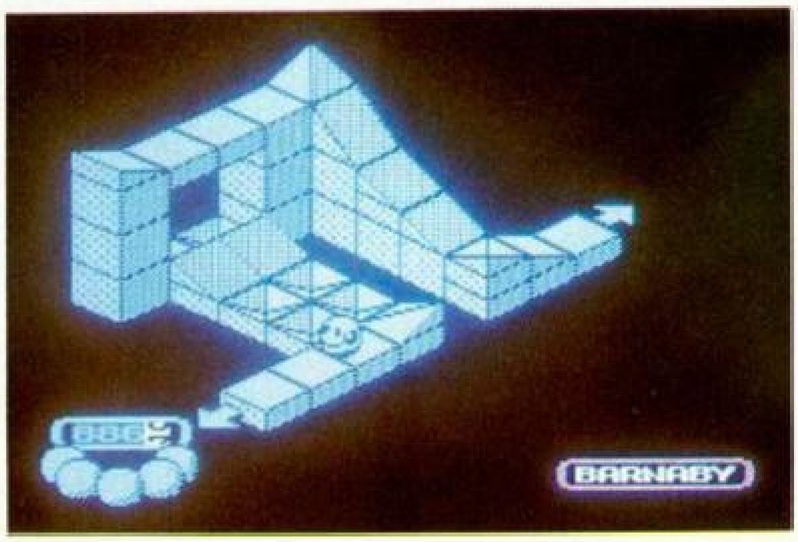


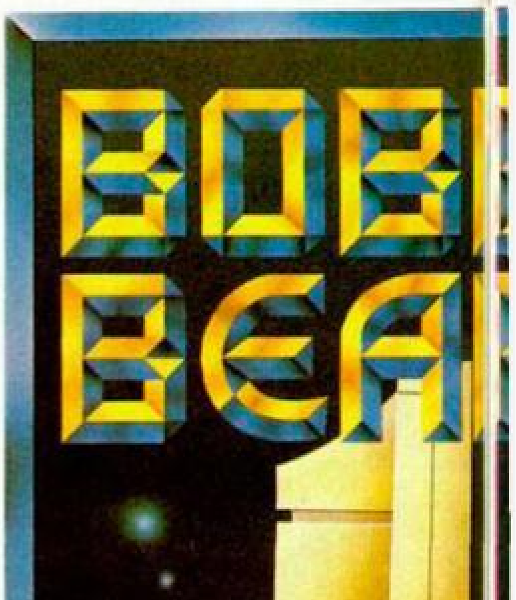
¡NUEVO!

BOBBY BEARING • Arcade • The Edge

EL PLANETA DE ACERO

Bobby Bearing es el último título editado por la prestigiosa compañía The Edge. A ella pertenecen títulos tan conocidos y brillantes como «That's the Spirit» o el sensacional «Fairlight». Estos antecedentes representan, sin duda, una excelente garantía de que el juego ante el que nos encontramos va a tener, cuanto menos, un mínimo nivel de calidad.





En efecto. El mero hecho de que este Bobby Bearing lleve en su carátula impreso el sello The Edge ya es suficiente como para sentarse cómodamente ante el ordenador y empezar a analizar detenidamente el juego que tenemos ante nosotros.

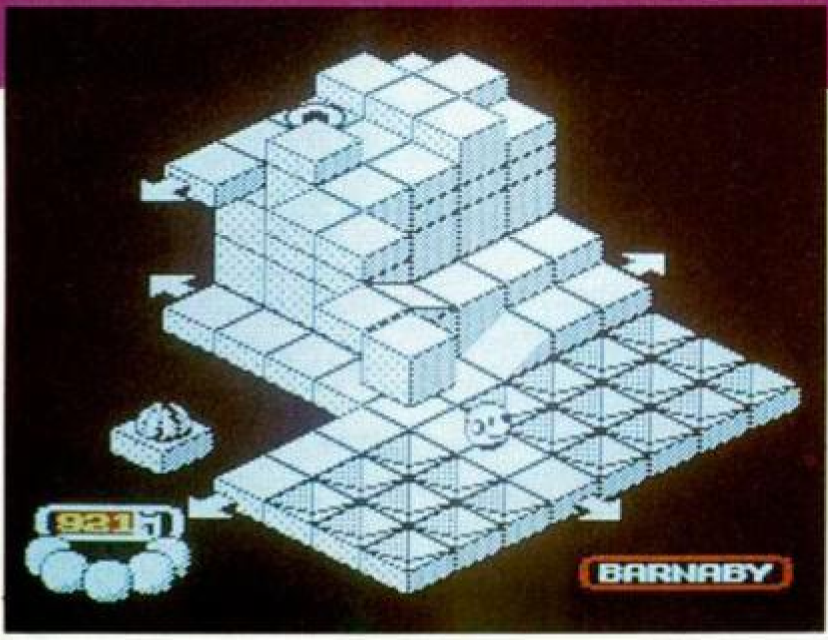
De entrada, así a primera vista, nos damos cuenta (a no ser que tengamos un serio problema visual), de que Bobby Bearing pertenece al prototipo de arcade que se desarrolla en un laberinto. Poco original, pensaréis. Pues no; estáis equivocados. Efectivamente, el escenario en el que tiene lugar el juego no es nada nuevo para nadie, pero la forma en que debemos actuar en él sí difiere sensiblemente de otros programas como Spindizzy, Gyroscope, etc...

Pero comencemos por comentaros brevemente la historia que rodea a Bobby Bearing. Este orondo y rápido mozalbete vive en Technofear en compañía de su familia. Este Technofear es un extraño y peculiar planeta en el que todo en él es de acero, incluso sus habitantes. Bobby ha recibido la visita de su primo, quien ha salido a jugar acompañado de los cuatro hermanos de Bobby. Estos han sido raptados por el malo de turno y ahora se nos plantea el problema de encontrar uno a uno a todos sus hermanos y hacerlos regresar a casa; luego ya veremos si podemos hacer lo propio con el primo...



Este es básicamente el argumento del programa, el cual, si bien resulta fácil en su concepción, se dificulta enormemente a la hora de su realización.

Bobby Bearing es fundamentalmente un juego de habilidad, aunque también es necesario para lograr un buen dominio del personaje el ser poseedores de un estupendo sentido de la orientación. Los laberintos que componen las pantallas del programa son verdaderamente complicados, no sólo por su propia estructura, sino por la gran similitud que guardan unas con otras. Estos suelen ser, en su gran mayoría, rela-

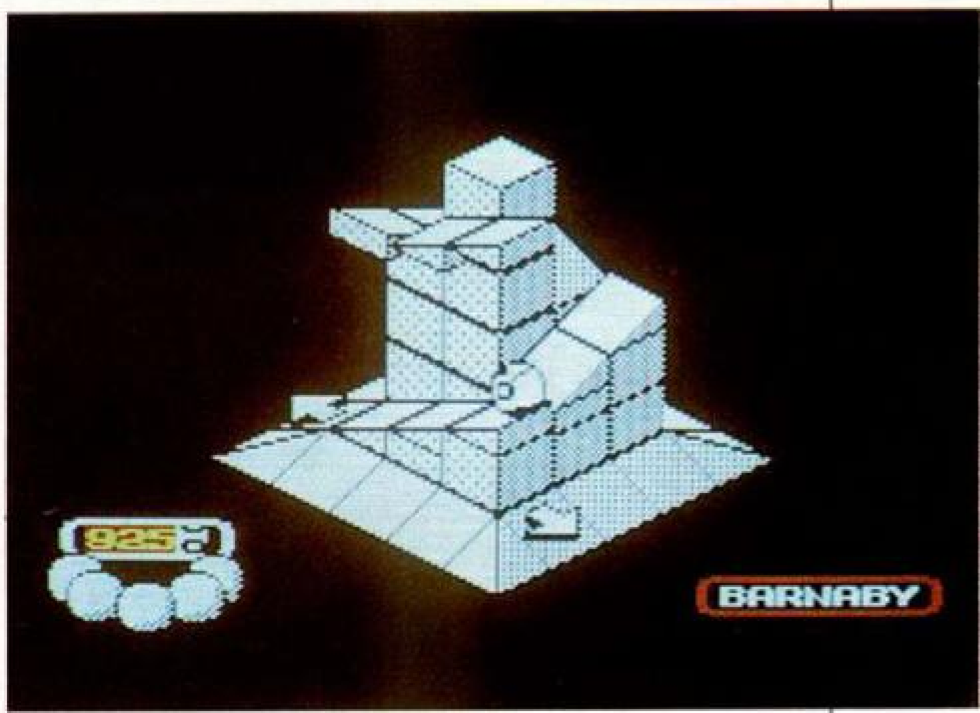


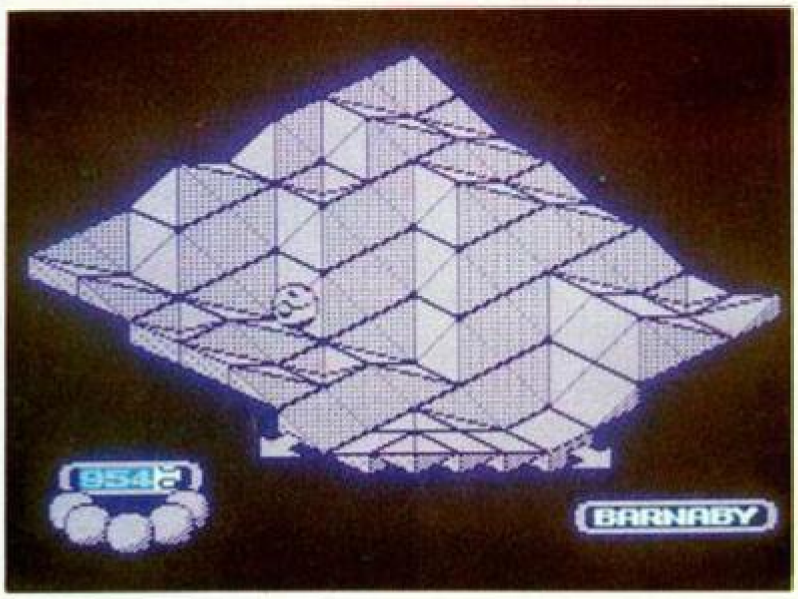


tivamente fáciles de recorrer y se puede andar por ellos a una gran velocidad, pero al poseer muchos de los pasillos idéntica estructura, diferenciada tan sólo por los colores de los mismos, hay que permanecer en todo momento con suma atención para tener conocimiento de dónde nos encontramos exactamente.

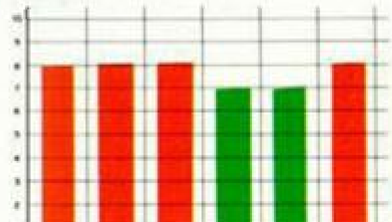
La misión del juego consiste, como ya sabéis, en encontrar a nuestros hermanos y devolverlos sanos y salvos a casa. Aquí es donde verdaderamente reside la dificultad y gracia del asunto. Al principio puede parecer un tanto simple el ir recorriendo las pantallas sin excesiva dificultad, pues, a diferencia de lo que ocurre en otros programas de este tipo, es imposible que nos salgamos de los laberintos y los únicos impedimentos existentes son los propios obstáculos del mismo. Pero cuando conseguimos localizar a alguno de nuestros pequeños familiares es cuando la cosa comienza a ponerse verdaderamente interesante. Nuestros hermanillos se encuentran aturdidos por el shock sufrido, por lo que no pueden moverse por sí solos. Por eso tendremos que ir empujándoles todo el camino de vuelta a casa. Si antes os parecía que Bobby Bearing era un juego sencillo, seguro que cambiáis rápidamente de opinión, pues el ir remolcando a un objeto inerte por estos pasadizos, recovecos y planos inclinados, resulta de lo más complicado, aunque también de lo más adictivo.

En efecto, Bobby Bearing es un programa muy bien pensado y mejor llevado a cabo. La habilidad y la inteligencia serán, en su justa medida, nuestras mejores armas para luchar contra el aburrimiento y poder disfrutar por entero de lo que este bonito y adictivo programa nos ofrece.





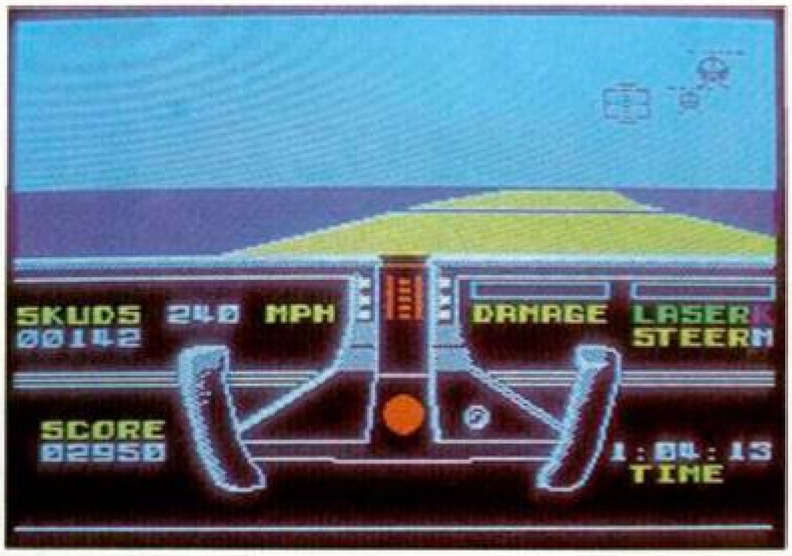
Por último, simplemente decir que el juego está compuesto por un número muy grande de pantallas, pero que esto es algo explicable y muy lógico si pensamos en que una parte muy elevada de las mismas son idénticas. Esta estratagema no quita, sin embargo, ningún mérito a sus programadores, quienes, desde luego, han llevado a cabo una gran labor de diseño.



KNIGHT RIDER • Arcade • Ocean

EL COCHE FANTASTICO

Knight Rider, o el Coche Fantástico como se le conoce en nuestro país, es un programa que no necesita presentación previa. De todos son conocidas las increíbles características de tan particular automóvil y el arrojo y valentía de su afamado conductor. Pero lo que posiblemente no sepáis es que la versión para ordenador realizada por Ocean, ha resultado, a nuestro juicio, un desastre.



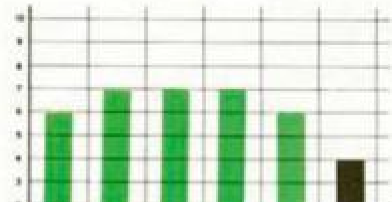
Pocas veces se volverá a presentar a los programadores una ocasión como ésta en la que el argumento de una popular serie de televisión se preste tanto para ser llevada al ordenador, (no olvidemos que el propio protagonista es un ordenador). Pero lamentablemente no se ha sabido o no se ha podido aprovechar la oportunidad y se ha llevado a cabo un programa que, calificándole por lo alto, resulta a lo sumo mediocre.

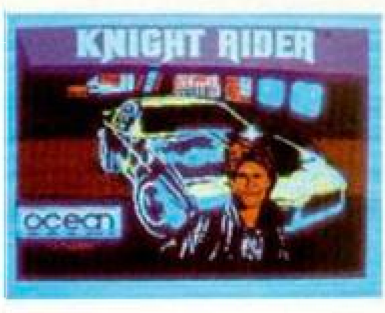


El juego consiste en esperar a recibir instrucciones y dirigirnos en nuestro particular vehículo hacia la ciudad de USA en la que ocurra el percance. Para ello tenemos dos posibilidades diferentes: llevar nosotros el láser mientras que Kit viaja automáticamente o conducir nosotros mientras él da buena cuenta de nuestros aéreos enemigos. Después de esta fase, nos introduciremos sigilosamente en el lugar de los hechos y, esquivando a los guardianes, deberemos hacernos con el objetivo deseado.

La primera fase resulta casi pésima, pues cualquier simulador deja a la altura del betún a lo que sucede en este Knight Rider. Tanto la realización de la carretera como el efecto de velocidad son verdaderamente malos. Si por el contrario preferimos disparar contra los helicópteros es posible que encontremos algo más de diversión, pero seguramente en escasas cantidades. La última parte del juego puede que sea un poco más emocionante, pero como siempre tenemos que hacer prácticamente lo mismo, acaba por aburrir.

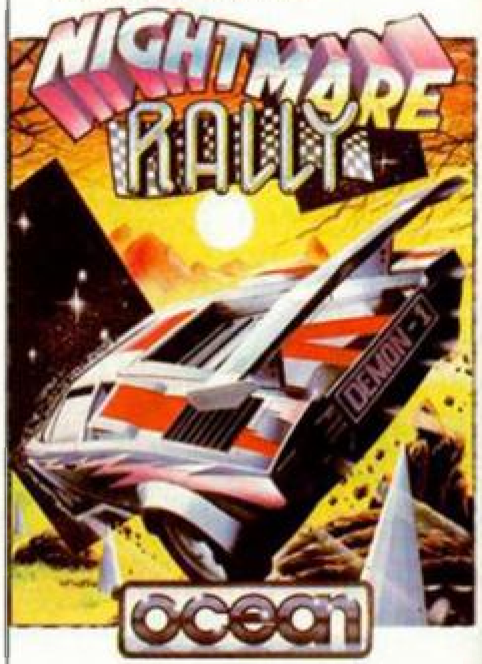
En definitiva, un programa pobre, gráficamente aceptable pero sumamente aburrido.





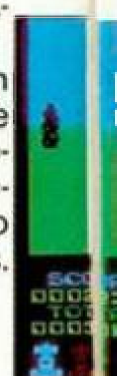
NIGHTMARE RALLY •

EMOCION A LAS CUATRO RUEDAS



Un original y emocionante simulador de coches nos llega de la mano de Ocean. Lo original reside en que hasta ahora prácticamente la totalidad de programas de este tipo nos ponían al mando de un fórmula 1 o cualquier otro tipo de automóvil supraveloz y nos situaban en la línea de salida de un asfaltado y curvado circuito de carreras. Sin embargo, nada de esto ocurre en Nightmare Rally; bueno, mejor dicho, casi nada, puesto que nuestro bólido también es francamente rápido, pero la diferencia estriba en que deberemos maniobrar por entre las piedras, árboles y obstáculos en general que se nos puedan presentar en un viaje campo a través.

Nightmare Rally es un juego verdaderamente divertido y muy emocionante debido a lo abrupto de los recorridos y lo irregular de sus trazados.

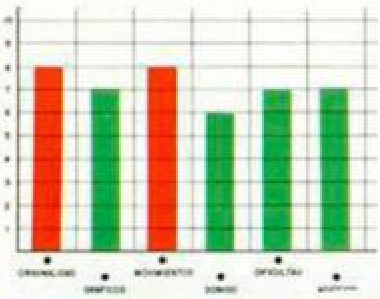


Simulador • Ocean

Este «rally de pesadilla» está formado, además, por varias pruebas diferentes en las que varían las circunstancias y el desarrollo de la conducción. Cada una de ellas es un buen test para demostrar nuestras grandes dotes como conductores y, a medida que vamos logrando completar los recorridos en una frenética carrera contra el reloj, nos vamos dando cuenta de que estamos ante un juego de lo más adictivo *(sobre todo si eres un amante de la velocidad y el peligro).*

Gráficamente, y en lo relativo a la credibilidad del juego *(es decir, si verdaderamente te crees o no que vas sobre un coche)* tenemos que decir que se ha conseguido un gran efecto, rápido y realista, pero sobre todo muy espectacular, debido a que podemos ver las acrobacias que nuestro propio vehículo realiza, ya que la visión de la pantalla nos incluye también a nosotros.

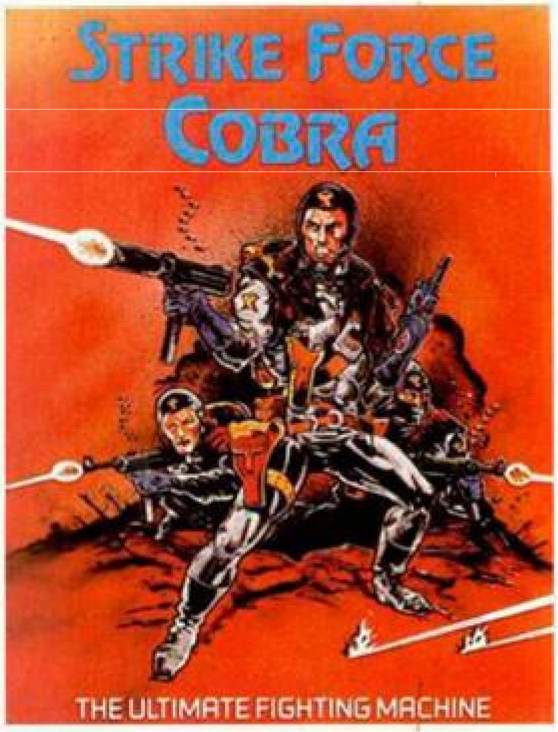
Si te gustan los simuladores y quieres descubrir algo nuevo, apúntate a este Nightmare Rally.





STRIKE FORCE COBRA • Videoaventura • Piranha

COMANDO DE CHOQUE



Una complicada misión se nos ha encomendado. El gobierno de la nación ha tenido noticias de que un «hacker» profesional ha conseguido un programa cargador con el que desproteger todo el sistema de ordenadores del gobierno. En él se encuentran todos los datos imprescindibles para el buen funcionamiento de una nación. Si este pirata monstruoso lograra salirse con la suya, no sólo podría convertirse en uno de los hombres más ricos del planeta, sino también de los más envidiados y, en su escalada, arrastraría al país hacia la más misera de las ruinas.



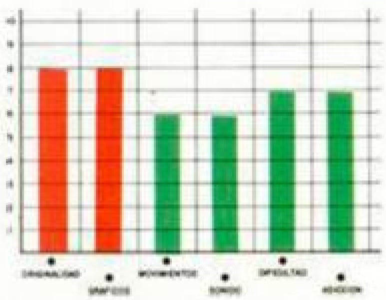
Hemos sido elegidos como el responsable de un comando cuyo único objetivo será el de acabar con el ordenador del genio malvado. Por eso se ha puesto a nuestra disposición a un nutrido grupo de personas, especialistas en todos los temas relacionados con el espionaje o la guerrilla, de entre cuyos miembros tendremos que seleccionar aquéllos que creamos reúnen las características adecuadas para finalizar con éxito tan heroica misión.

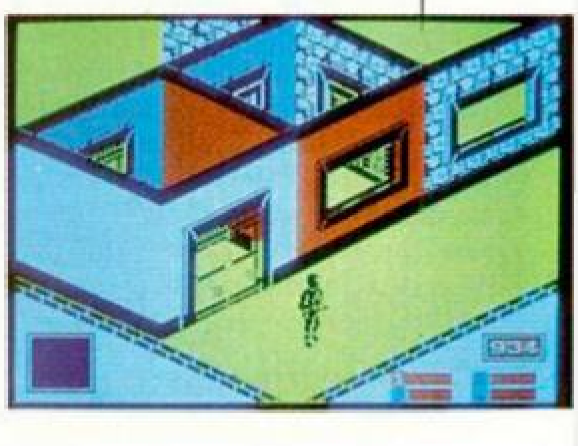
Tendremos que conseguir que cada uno de ellos cumpla su parte de cometido y encuentre la clave correspondiente para permitir que sus compañeros puedan continuar con la misión. Si uno falla, el trabajo de los demás habrá sido en vano; por eso se exige de cada uno de ellos un máximo de eficacia.

Como habréis podido imaginar, Strike Force Cobra es un juego básicamente de estrategia, pero otros factores como la habilidad y la fuerza van a contribuir en el resultado final de la misión. Además de tener que plantearnos una estrategia correcta, necesitamos que luego ésta sea llevada a la práctica. Por eso será necesario enfrentarnos a múltiples guardianes o conseguir información de algunas terminales.

Si te atreves a afrontar esta misión obtendrás como resultado unas buenas horas de diversión, mientras que disfrutas de los excelentes diseños y bellas pantallas por las que transcurre el argumento de este Strike Force Cobra.

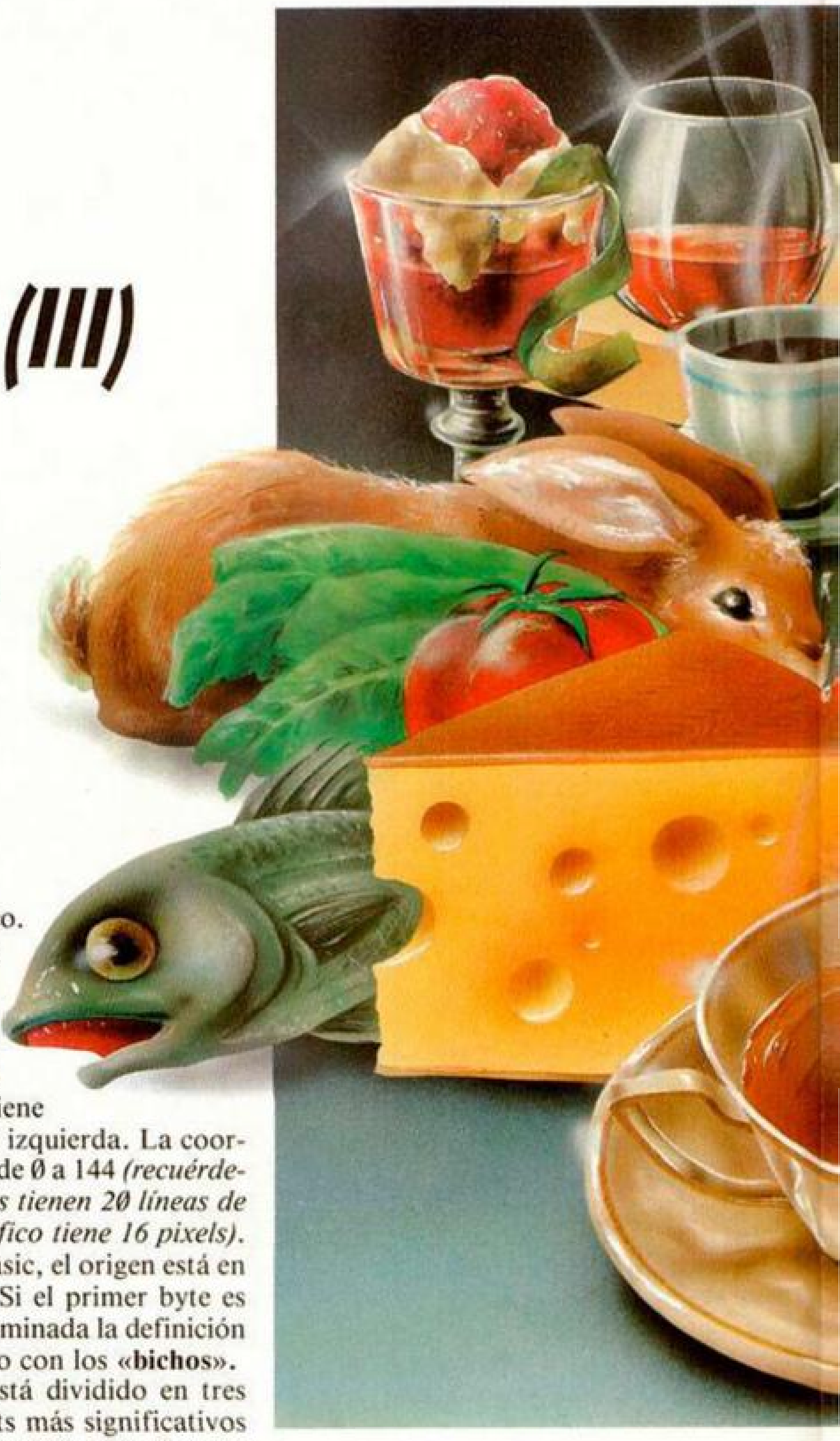
Un buen juego y bastante completo. De esos que podemos pasarnos meses para llegar a completar con éxito, pero que divierten.





APRENDE A PROGRAMAR TU PROPIO JUEGO (III)

Pablo Ariza



En el capítulo de hoy terminaremos la explicación del listado publicado la semana pasada, y además, podremos regalarnos los oídos con la música que servirá de acompañamiento al juego.

Al definir una pantalla, además de la forma de ésta, tema que tratamos la semana pasada, hay que consignar cuáles son los «**bichos**» que se encuentran en ella. Antes de pasar a ver cómo están almacenados los datos de estos simpáticos personajes en la memoria, vamos a hacer una serie de puntualizaciones:

— Todos los «**bichos**» serán gráficos de 16×16 pixels con un solo color.

— Algunos serán animados *(tendrán varios dibujos distintos que al alternarlos darán la sensación de animación)*. Si lo son, pueden tener dos o tres fases de animación.

— Harán siempre un recorrido constante. Irán en cualquier dirección en línea recta, haciendo recorrido de ida y vuelta.

— Algunos serán «**direccionales**», es decir, su gráfico será distinto cuando van en un sentido que cuando van en el otro.

Ahora vamos a ver cómo organizamos los datos de estos «**bichos**» móviles. Como recordaréis, irán colocados a continuación de los datos que definen los bloques de decorados, y separados por un 255.

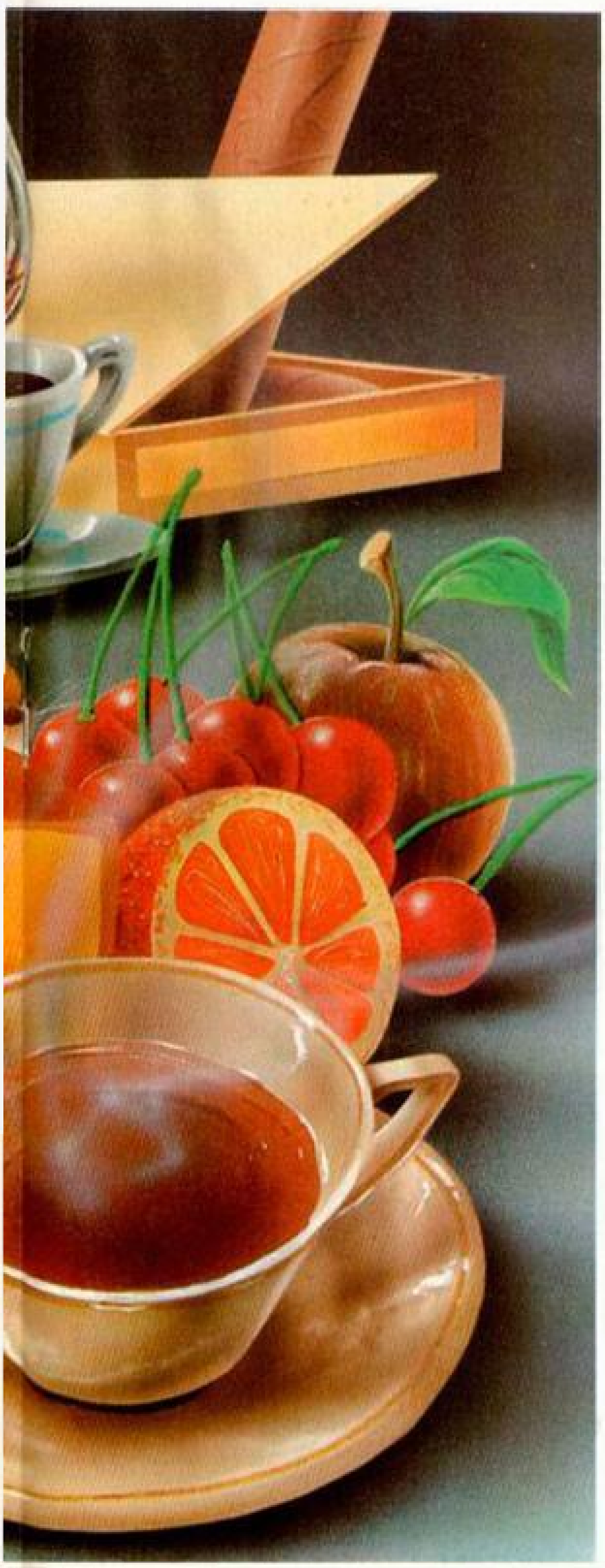
Para cada uno de los «**bichos**» existentes en la pantalla, necesitaremos cinco bytes. Su significado lo tenéis resumido en la **figura 1**. Los bytes primero y segundo nos dan las coordenadas donde comienza el movimiento del «**bicho**», pero esta vez en alta resolución. Se dan las coordenadas de la esquina superior izquierda del gráfico. La coordenada X puede ir de Ø a 24Ø *(si tomara un valor mayor, el «**bicho**» se saldría de la pantalla)*, y tiene por origen la parte izquierda. La coordenada Y puede ir de Ø a 144 *(recuérdese que las pantallas tienen 2Ø líneas de altura y que el gráfico tiene 16 pixels)*. A diferencia del Basic, el origen está en la parte superior. Si el primer byte es 255, se dará por terminada la definición de la pantalla junto con los «**bichos**».

El tercer byte está dividido en tres partes. Los dos bits más significativos guardan la velocidad. Este parámetro puede valer 1, 2 ó 3. Cuanto mayor es, más lento se mueve el «**bicho**». Los otros seis bits están divididos en dos grupos de tres bits que indican los incrementos X *(horizontal)* e Y *(vertical)*. Estos son los valores que se van sumando a las coordenadas iniciales. Ambos pueden ir de —4 a +3. Dependiendo de la dirección que queramos que tome nuestro «**bicho**», deberemos hacer una u otra combinación de incrementos. Además, también influyen en la velocidad. Cuanto mayores sean *(en valor absoluto)*, más rápidamente se moverá el «**bicho**», pero su movimiento será también más brusco, puesto que se realizará de más en más pixels.

El cuarto byte indica con sus tres bits superiores cuál es el color del «**bicho**», y con sus otros cinco, cuál es su código. El código servirá después para saber qué figura le corresponde, si es animado, si es direccional, etc.

Por último, el quinto byte indica la longitud del recorrido. Por ejemplo, si la longitud es 5 el incremento X es 1 y el incremento Y es Ø, el «**bicho**» se moverá hacia la derecha 5 pixels y luego volverá 5 pixels hacia la izquierda, repitiendo después el ciclo.

Como ya dijimos antes, usando el código, podemos conocer otras características del **bicho**. Para ello necesitaremos de una tabla a la que accederemos usando el código como contador. Cada elemento de esta tabla constará de cinco bytes. Su significado está en la **figura 2.**



Los dos primeros indican la dirección de las figuras hacia la izquierda y los dos siguientes hacia la derecha. Esto sirve para los **«bichos»** direccionales, cuyas figuras son distintas en un sentido que en otro. Si el **«bicho»** no es direccional, entonces los dos primeros bytes serán iguales a los dos siguientes *(el primero igual al tercero y el segundo igual al cuarto)*. Lo que indica cada grupo de dos bytes es la dirección de memoria donde está ubicado el gráfico correspondiente al **«bicho»** en cuestión. Si es animado, los distintos graficos estarán colocados uno detrás de otro empezando en la dirección indicada, con lo que sólo necesitamos saber en qué parte de la memoria está el primero para saber dónde están los demás.

PROGRAMA 1

```
  10 PAPER 0: INK 7: BORDER 0: C
LEAR 32767: LOAD "cm3 1"CODE 611
83,158: LOAD "cm3 2"CODE 50432,1
988
  20 FOR X=40000 TO 40029: READ
A: POKE X,A: NEXT X
  30 DATA 42,118,92,34,1,239,34,
3,239,62,238,237,71,237,94,175,2
19,254,246,224,60,40,248,62,63,2
37,86,237,71,201
  40 INPUT A: RANDOMIZE A
  50 IF INKEY$<>"" THEN GO TO 50
  60 RANDOMIZE USR 40000: GO TO
40
```

El quinto y último byte indica cuántas fases de animación tiene. Si no tiene animación, se indicará como una fase *(un solo gráfico distinto)*.

Cada vez que entramos en una pantalla, utilizando los datos que vienen en la definición de ésta y los obtenidos en la tabla que acabamos de explicar, hay que crear una tabla de trabajo en la que se incluirán además otros datos necesarios durante nuestra estancia en la pantalla para mover los **«bichos»**. Esta es la única tabla que se utilizará a la hora de moverlos, las demás sólo sirven para crear ésta. Cada vez que se sale de una pantalla, la tabla de trabajo es destruida, creándose en su lugar la tabla de trabajo de la nueva pantalla donde estamos.

Esta tabla consta de 16 bytes por cada **«bicho»**. Podéis encontrarlos en la **figura 3**. Los dos primeros bytes indican las coordenadas del **«bicho»**. Comienzan siendo iguales que los dos primeros bytes de los cinco que definen cada **«bicho»** en los datos de la pantalla, pero luego van variando según el **«bicho»** se va moviendo. Los bytes tercero y cuarto indican los incrementos X e Y, que comienzan también siendo iguales que los consignados en los datos de pantallas, y que, luego, son periódicamente cambiados de signo para conseguir que el movimiento se realice en un sentido y en el otro.

LISTADO 1

Línea	Datos	Control
1	06EF42C500C500F3F5E5	1422
2	D5C5DDE508F53A05EF3C	1475
3	E6033205EF20172A01EF	864
4	5E1C20052A03EF5E1C1D	594
5	23E5CD3CEFE1232201EF	1302
6	F108DDE1C1D1E1F1FBED	2051
7	4D6EAF6757016BEF0966	1010
8	6CEB09666C01070008AF	753
9	000008EB25D3FE200AEE	1025
10	106510F20DC251EFC9C6	1301
11	0010E90DC251EFC9D4C8	1389
12	BD83A99F968E867E7770	1479
13	6A645F5954504B47433F	830
14	3C3835322F2D2A282623	466
15	21201E1C1B1918161514	262
16	131211100F0E0D010000	113

BLOQUE: cm3 1
DUMP: 40000
N.° BYTES: 158

Los bytes quinto y sexto indican la dirección de memoria donde se encuentra el gráfico que fue dibujado en pantalla la última vez. Necesitamos este dato porque, como después veremos, el dibujo de las figuras lo haremos con el método del OVER 1, con lo que cada vez que el **«bicho»** se mueva, deberemos dibujar la nueva figura y borrar la anterior, lo que se hace dibujándola encima otra vez.

El séptimo byte es el contador de la animación. Va incrementándose progresivamente, y sirve para saber en cada momento cuál de las varias figuras que tienen los **«bichos»** animados es la que debe dibujarse a continuación.

Los bytes octavo y noveno, y los bytes décimo y undécimo indican las direcciones de memoria de los gráficos del **«bicho»** a izquierda y a derecha. Son tomadas directamente de la tabla de **«bichos»** de la que hablamos antes.

El duodécimo byte tiene dos funciones. Sus tres bits superiores indican el color. El resto de los bits indican el número de fases de la animación.

DATOS DE LOS «BICHOS» EN LA DEFINICION DE PANTALLAS

Byte 1	**Coordenada X (0-240)**
Byte 2	**Coordenada Y (0-144)**
Byte 3	**— Bits (0-2, incremento Y** **— Bits 3-5, incremento X** **— Bits 6-7, velocidad**
Byte 4	**— Bits 0-4, código** **— Bits 5-7, color**
Byte 5	**Longitud del recorrido**

Figura 1.

El byte decimotercero es el contador de la longitud, y el decimocuarto indica la longitud. El contador va siendo incrementado, y cuando es igual a la longitud se pone nuevamente a cero y se cambia el signo de los incrementos para que el **«bicho» se dé la vuelta**.

El byte decimoquinto es el contador de la velocidad y el decimosexto y último indica la velocidad. Al igual que antes, el contador es incrementado cada vez. Cuando alcanza a la velocidad, se vuelve a poner a cero, y sólo en ese momento es cuando se mueve el **«bicho»**.

Por eso, cuanto mayor sea el número de la velocidad, mayor será el tiempo que tarde el contador en alcanzarla, y más lento será el movimiento. Quizá

LISTADO 2

```
  1  180C180C1831180C1C0B  220
  2  1C0B1C311C0B1A091A09  225
  3  1A311A09180718071831  245
  4  1807170E170E1731170E  214
  5  130B130B1331130B1509  188
  6  15091531150917071707  190
  7  17311707180C180C3131  272
  8  31131F101F13310B310B  285
  9  310C310C313131131C10  332
 10  1C0E1C0C1A0C180C180C  192
 11  31313107130913073110  273
 12  3110310C310C31313107  341
 13  180C180B18091509170B  168
 14  170B17311A0713091307  193
 15  310C310C310B310B3131  340
 16  31071509150715041702  164
 17  180C180C18311C071309  208
 18  130731103110310C310C  278
 19  31313107150915071510  249
 20  1710180C180C31313113  277
 21  1F101F13310B310B310C  278
 22  310C313131131C101C0E  313
 23  1C0C1A0C180C180C3131  248
 24  31071309130731103110  240
 25  310C310C31313107180C  312
 26  180B18091509170B170B  166
 27  17311A0713091307310C  220
 28  310C310B310B31313107  335
 29  150915071510170E180C  168
 30  180C183113071C091C07  207
 31  1C101310180E18313131  288
 32  310E18311831310C310C  331
 33  FF240C240C2431243128  561
 34  10281026312631263126  371
 35  31240C240C240C240C23  276
 36  10231023312331243124  356
 37  31241024101F101F101F  278
 38  311F3118101810181018  273
 39  101D111D111D311D311D  293
 40  311D311D311D311D311D  390
 41  311D311D311D111D1131  346
 42  1131111F131F13213121  298
 43  311F311F311F131F1321  342
 44  31213121112111231323  320
 45  13233123312331233124  391
 46  0C240C24312431281028  326
 47  10260C260C260C260C24  252
 48  102410240C240C281028  266
 49  10281028102931293129  355
 50  0C290C28102810283128  306
 51  31260026002410241024  265
 52  11241129312931293129  381
 53  31293129312931293129  450
 54  31293129112911293129  386
 55  3128112B112D132D132B  340
 56  132B132B132B132D132D  314
 57  132D132D132F132F132F  326
 58  312F312F312F31240C24  421
 59  0C240C240C280C280C26  250
 60  0C260C260C260C240C24  246
 61  0C240C240C240C240C24  240
 62  10241024102410281028  266
 63  10261026102610261024  268
 64  10241024102410241024  260
 65  10241124112411241128  268
 66  11261126112611261126  277
 67  1124112411241124112B  272
 68  112B112B132B132B132B  306
 69  13291329132813281328  297
 70  13281326132613261326  287
 71  1326132613FF1F131F13  488
 72  310E310E310E310E1A0E  292
 73  1A0E21152115310E310E  274
 74  310E310E1A0E1A0E2313  260
 75  23132315231523172317  282
 76  23182318311A311A1A31  343
 77  1A311C311C311D311D31  385
 78  1E311E311E121E122131  336
 79  2131211221121A311A31  334
 80  1A121A121C101C101E0E  220
 81  1E0E1F311F311F131F13  304
 82  23312331231323131A31  351
 83  1A311A131A131A101A10  249
 84  1A0E1A0E1F311F313113  308
 85  3113310E310E1C311C31  348
 86  23312331311031103117  370
 87  31171C311C3124152415  340
 88  2417241724182418241A  300
 89  241A311C311C1E311E31  374
 90  1C311C311B311B311A0E  346
 91  1A0E1A101A101C111C11  214
 92  1C0E1C0E1E1A1E1A1E18  250
 93  1E181A171A171A151A15  246
 94  1F131F131F171F171F0E  253
 95  1F0E1F171F1731133113  289
 96  1A311A311C311C311D31  382
 97  1D311E311E311E121E12  332
 98  21312112211221311A12  310
 99  1A121A311A121C121C10  253
100  1E0E1E0E1F311F311F13  298
101  1F132331231323132331  326
102  1A131A131A311A131A13  255
103  1A101A0E1A0E1E311E31  280
104  1E121E12213121122112  280
105  21311A121A121A311A12  289
106  1C121C101E0E1E0E1F31  258
107  1F311F131F1323312313  318
108  231323311A131A131A31  303
109  1A131A131A101A0E1A0E  212
110  20142014203120312331  350
111  2314231423141C101C10  253
112  1C311C311E311E102010  327
113  20102115211521312131  320
114  24312415241524151C10  300
115  1C101C311C311C311C10  319
116  1C101C101E0E1E0E1E0E  220
117  1E0E210E210E210E210E  232
118  1A101A101A101A101C10  212
119  1C101E101E1021122112  238
120  211221121A121A121A12  234
121  1A121C151C151C151C15  240
122  1E151E151E151E15FF18  483
123  3118311A311A311C0C1C  340
124  0C1C311C311C101C101D  283
125  311D311F101F10213121  336
126  311C0C1C0C1C0C1C0C1C  237
127  101C101C311C31183118  311
128  311A311A311C0C1C0C1C  307
129  311C311C101C101D311D  321
130  311F101F102131213115  328
131  09150915091509150C15  153
132  0C153115311831183117  321
133  31173113071307130713  218
134  07130513051305130515  124
135  04150417041704180918  140
136  09180918091807180718  161
137  07180717071707170717  151
138  07150515051505150515  132
139  07150715071507130713  136
140  07130713071015101510  149
141  15101510131013101310  179
142  13183118311A311A311C  343
143  0C1C0C1C311C311C101C  278
144  101D311D311F101F1021  299
145  3121311C0C1C0C1C0C1C  279
146  0C1C101C101C311C3118  278
147  3118311A311A311C0C1C  340
148  0C1C311C311C101C101D  283
149  311D311F101F10213121  336
150  31150915091509150915  190
151  0C150C15311531183118  282
152  31173117311307130713  264
153  07130713051305130513  124
154  05150415041704170418  133
155  0C180C18311831180C18  254
156  0C170E170E1510151015  181
157  101510170B170B170B17  178
158  0B170717071507150713  146
159  07130713071307150C15  139
160  0C150C150C150C150C15  165
161  0C150C15311531173117  280
162  31180918091809180918  205
163  09180917311731180718  241
164  07183118311A0B1A0B1A  253
165  0B1A0B1A0B1A0B1A311A  223
166  31180B180B1A311A311C  297
167  0C1C0C1C0C1C0C1C0C1C  200
168  0C1A0C1A0C1C0B1C0B1C  194
169  0B1C0B13051305130513  141
170  05130513051305130515  122
171  31153113311331110511  294
172  05113111311105110513  200
173  07130715091509150915  144
174  09130B130B1331133113  224
175  07130713311331150415  215
176  04133113311000100010  188
177  31103110001000110211  182
178  02130413041304130411  111
179  09110911311131110711  208
180  07113111311505150517  214
181  311731180C180C310C31  303
182  0C183118311A311A311C  336
183  0B1C0B1C0C1C0C170E17  190
184  0E170E170E1C131C131C  210
185  111C111A131A131A131A  223
186  13180718071807180717  166
187  0C170C180C180C130B13  168
188  0B130C130C1009100910  139
189  09100915091509150715  143
190  07130513051305130511  120
191  0C110C110B110B110911  140
192  09130913091507150715  142
193  051505130B130B130B13  140
194  0B150915091509150917  154
195  07170717071707150915  148
196  09150915091509150915  150
197  09150915091509150915  150
198  09150915091509150915  150
199  091509150E1A0EFF0000  369
```

fuera más apropiado llamarlo «**antivelocidad**».

El programa que, utilizando estos datos, se encarga de mover los «**bichos**», lo veremos próximamente. Ahora vamos a ver el programa que crea estos datos a partir de los datos de la definición de la pantalla. Puede ser algo complicado de entender, porque se trabaja con tres tablas a la vez. Antes hay que dejar claro algo con respecto a la nomenclatura. Cuando se hable de «**datos de bichos**», nos estaremos refiriendo a la primera de las tablas explicadas hoy, y que es donde se encuentran los datos que nos dicen qué «**bichos**» hay en una determinada pantalla. De este tipo de tabla, hay una por cada pantalla. Cuando se hable de «**tabla de bichos**», nos estaremos refiriendo a la segunda de las tablas explicadas hoy, a la que se accede mediante el código del «**bicho**». De esta tabla sólo hay una, ya que recoge todos los posibles «**bichos**» que puedan aparecer en cualquier pantalla. Por último, cuando se hable de «**tabla de trabajo**», nos estaremos refiriendo a la última tabla que hemos explicado, la utilizada después por la rutina de movimiento de los «**bichos**», y la que tiene que crear el programa que vamos a explicar a continuación.

El listado

El listado de este programa se publicó la semana pasada y está a partir de la etiqueta DIBI. Una vez aclarado esto, procedamos a la explicación.

Por si no lo recordáis, a la etiqueta DIBI se llegaba desde la rutina que dibuja la pantalla cuando en la dirección apuntada por DE había un 255. Al incrementar DE, éste señala el principio de los datos de los «**bichos**». Se guarda DE momentáneamente en la pila mientras se borran 100 bytes a partir de la dirección 23296. Es aquí donde se creará la tabla de trabajo. Se recupera DE y se hace que HL apunte al comienzo de la aún no creada tabla de trabajo. En BIBILO comienza el bucle para cada uno de los «**bichos**». En primer lugar, traslada la coordenada X de los datos de los «**bichos**» a la tabla de trabajo. Si es 255 retorna al punto desde donde fue llamada, pues ya están creados todos los datos que se necesitarán posteriormente durante nuestra estancia en la pantalla. Recuérdese que el punto de entrada había sido DESCOM. De no ser así, traslada también la coordenada Y.

Toma después de (DE) el tercer byte de los datos de «**bichos**». De éste pasa a B los dos bits superiores *(la velocidad)*. Toma entonces los 3 bits correspondien-Guardamos DE en la pila porque lo vamos a necesitar para otra cosa. Pasamos entonces a DE el puntero de la tabla de trabajo. Multiplicamos el código por

TABLA DE LOS «BICHOS»

Bytes 1 y 2	Dirección de los gráficos a izquierda
Bytes 3 y 4	Dirección de los gráficos a derecha
Byte 5	Número de fases de la animación

Figura 2

tes al incremento X, y mediante el uso de SRA A los pasa a los tres bits inferiores de A, cargándolo luego en la tabla de trabajo. Acto seguido, hacemos lo mismo con el incremento Y. En los bytes quinto y sexto metemos los valores 1ØØ Y 91 respectivamente. Estos bytes indicaban la dirección de la última figura dibujada en la pantalla, pero como aún no ha sido dibujada ninguna metemos la dirección 23396, en donde se encuentra un gráfico en blanco. Al hacer después dos INC HL seguidos nos saltamos el contador de la animación porque su valor inicial debe ser Ø. Ahora tomamos de *(DE)* el código del **bicho.** cinco y se los sumamos a la dirección donde comienza la tabla de «**bichos**». Como para esto necesitamos utilizar el registro BC, antes debemos guardar el contenido de B *(la velocidad)* en A. Con el LDIR de 4 bytes, pasamos las direcciones de las figuras de izquierda y derecha de la tabla de «**bichos**» a la tabla de trabajo. Volvemos a dejar ahora en B el valor de la velocidad. Cargamos en A el número de fases de la animación. Ya no necesitamos para nada la tabla de «**bichos**», así que recuperamos en HL el puntero de los datos de «**bichos**» que estaba señalando al tercer byte. Mezclamos los tres bits superiores de éste *(el color)*, con el contenido de A *(las fases de animación)*, y cargamos el resultado en la dirección indicada por DE *(el decimosegundo byte de la tabla de trabajo)*. Inicializamos ahora el contador de la longitud con 255 (por —1), pasamos la longitud del recorrido de los datos de «**bichos**» a la tabla de trabajo. Nos pasamos por alto el contador de la velocidad *(que debe valer Ø inicialmente)*, y cargamos en el último byte de la tabla de trabajo el valor de B, que era la velocidad. Antes, con EX DE, HL, hacemos que nuevamente HL sea puntero de la tabla de trabajo y DE lo sea de los datos de «**bichos**». Incrementamos DE y HL para que apunten al comienzo de los datos de los siguientes elementos y cerramos el bucle.

TABLA DE TRABAJO	
Byte 1	Coordenada X
Byte 2	Coordenada Y
Byte 3	Incremento X
Byte 4	Incremento Y
Bytes 5 y 6	Dirección última figura dibujada
Byte 7	Contador de animación
Bytes 8 y 9	Dirección de figuras izquierda
Bytes 10 y 11	Dirección de figuras derecha
Byte 12	— Bits Ø a 4, número de fases animación — Bits 5 a 7, color
Byte 13	Contador de la longitud
Byte 14	Longitud del recorrido
Byte 15	Contador de la velocidad
Byte 16	Velocidad

Figura 3.

La música

Hoy publicamos los listados que nos permitirán escuchar cuatro de las músicas que tendrá el juego. Son las cuatro melodías que servirán de fondo musical durante el juego. El resto de las músicas, que publicaremos en otro capítulo, serán para presentación final, lista de récords, etc. Cada cosa tiene su música. Para poder escucharlas, teclear en primer lugar el **programa 1** y grabarlo en cinta con LINE 1Ø. Teclear después, con el cargador universal de código máquina, los **listados 1 y 2**, haciendo DUMP en la dirección 4ØØØØ. La longitud del **listado 1** es de 158 bytes y la del **2**, de 1988. Grabarlos a continuación del BASIC. Primero el **listado 1**, llamándolo **«cm3 1»**, y después el **listado 2**, llamándolo **«cm3 2»**. Ahora ya se puede cargar todo con LOAD «». Al terminar la carga nos pedirá un número. En la **figura 4** están los números a introducir para cada una de las músicas. La melodía elegida sonará indefinidamente hasta que pulsemos una tecla, momento en que se nos pedirá otro número para escuchar otra música.

También publicamos en este capítulo el listado en ensamblador de la rutina de generación de música. Sin embargo, no se explicará en detalle por su excesiva complejidad, requiriéndose conocimientos que no los tiene mucha gente, y estos artículos intentan ser una iniciación al alcance de todo el mundo. El explicar todos estos temas, aun sin meternos en muchos detalles, podría llenar muchos capítulos. Además, al ser simulación de dos canales de sonido, el programa se complica aún más. A todos los que reúnan los conocimientos necesarios, no les resulta difícil entender el listado, aun sin explicación. De todos modos, haremos unas pequeñas indicaciones para todos los que estén interesados en ello:

— Mientras las interrupciones estén activadas, el registro I contendrá 238, con lo que el puntero de la interrupción estará en la dirección 61183 (238*256 + + 255).

— La música se almacena en la memoria en grupos de dos bytes, que indican las notas de cada canal. El final de la música se indica con un 255. Las notas van de Ø a 48, siendo el 24 el DO central. Los silencios se indican con 49.

— La variable CON indica cuántas interrupciones han pasado desde la última vez que se tocó una nota. Se toca una nota cada cuatro interrupciones (en realidad es cada cinco, ya que cuando se toca la nota, se sobrepasa el tiempo de una interrupción, con lo que se pierde una ídem).

— La variable SPAR contiene la dirección donde están las siguientes notas a tocar. La variable SPOR indica la dirección donde comienza la música.

— La subrutina que verdaderamente toca las dos notas simultáneamente es SOUNSUB.

— Para cada nota se necesita un dato, que está en DEFBs a partir de la etiqueta TABLA. Cuanto mayor es el número, menor es la frecuencia de la nota.

Con estas indicaciones podrá ser bastante más fácil la comprensión del listado, aunque, en realidad, esto no es necesario para la comprensión del resto del juego.

LISTADO ENSAMBLADOR

```
*C-
*D+
;
;RUTINA DE INTERRUPCIONES
;
        ORG  61183
;
        DEFW ENTER
SPAR    DEFW 0
SPOR    DEFW 0
CON     DEFB 0
ENTER   DI
        PUSH AF
        PUSH HL
        PUSH DE
        PUSH BC
        PUSH IX
        EX   AF,AF'
        PUSH AF
        LD   A,(CON)
        INC  A
        AND  3
        LD   (CON),A
        JR   NZ,EXDIR
        LD   HL,(SPAR)
        LD   E,(HL)
        INC  E
        JR   NZ,SIG
        LD   HL,(SPOR)
        LD   E,(HL)
        INC  E
SIG     DEC  E
        INC  HL
        PUSH HL
        CALL SOUN
        POP  HL
        INC  HL
        LD   (SPAR),HL
EXDIR   POP  AF
        EX   AF,AF'
        POP  IX
        POP  BC
        POP  DE
        POP  HL
        POP  AF
        EI
        RETI
;
;
SOUN    LD   L,(HL)
        XOR  A
        LD   H,A
        LD   D,A
        LD   BC,TABLA
        ADD  HL,BC
        LD   H,(HL)
        LD   L,H
        EX   DE,HL
        ADD  HL,BC
        LD   H,(HL)
        LD   L,H
        LD   BC,7
        EX   AF,AF'
        XOR  A
EXINT   NOP
        NOP
ININT   EX   AF,AF'
        EX   DE,HL
        DEC  H
        OUT  (254),A
        JR   NZ,DUPL
        XOR  16
        LD   H,L
        DJNZ EXINT
        DEC  C
        JP   NZ,ININT
        RET
DUPL    ADD  A,0
        DJNZ EXINT
        DEC  C
        JP   NZ,ININT
        RET
;
;
;    TABLA DE FRECUENCIAS
;
;
 TABLA  DEFB 212,200,189,17
             9,169,159,150
        DEFB 142,134,126,11
             9,112
        DEFB 106,100,95,89,
             84,80
        DEFB 75,71,67,63,60
             ,56
        DEFB 53,50,47,45,42
             ,40
        DEFB 38,35,33,32,30
             ,28
        DEFB 27,25,24,22,21
             ,20
        DEFB 19,18,17,16,15
             ,14,13,1
```

TABLA DE MUSICAS	
Número	**Música**
50432	1
50753	2
51138	3
51651	4

Figura 4.

MICRO Manía

Sólo para adictos

DINAMITE DAN

No. Aún no os vamos a contar nada de la segunda parte de tan maravilloso juego. De momento vais a tener que seguir sacándole partido a ese que tanto os gustó en su día y que seguro nunca llegasteis a completar. Tirando de nuevo de nuestro eficiente archivo comprobamos que en el número 90 yas os ofrecimos un interesante poke; pues bien, por si acaso hay por ahí algún despistadillo os lo volveremos a ofrecer, pero esta vez acompañado del correspondiente para obtener, ¿qué?, eso, sí, ¡vidas infinitas!

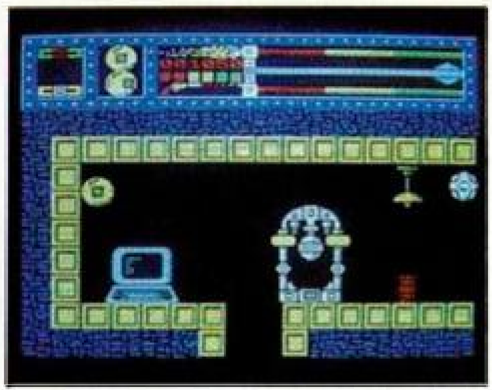
POKE 58770, 20 para eliminar enemigos.

POKE 52678, 0 para vidas infinitas.

P.D. Esta sección está nutrida esta semana gracias a la inestimable colaboración de David.

EQUINOX

¿Qué?, ¿ya se os están poniendo los dientes largos, eh? Está bien. No os haremos sufrir más. Seguro que estabais esperando como lobos ambrientos que os contáramos algún poke para este gran programa de Mikro-Gen. Pues ahí tenéis, ¡tomad carnaza y devorarla con fruición! POKE 49538, 62: POKE 49539, 2: POKE 49540, 50: POKE 49541, 255: POKE 49542, 197: POKE 49543, 0 y disfrutad de las suculentas vidas infinitas.



RASPUTIN

En esta misma sección del número 88 de nuestra revista, (para que veáis que nuestro sistema de archivo funciona a la perfección), os ofrecíamos un curioso truco para el juego de Firebird «Rasputín». Pues atentos que el de hoy es aún más asombroso e igualmente peculiar.

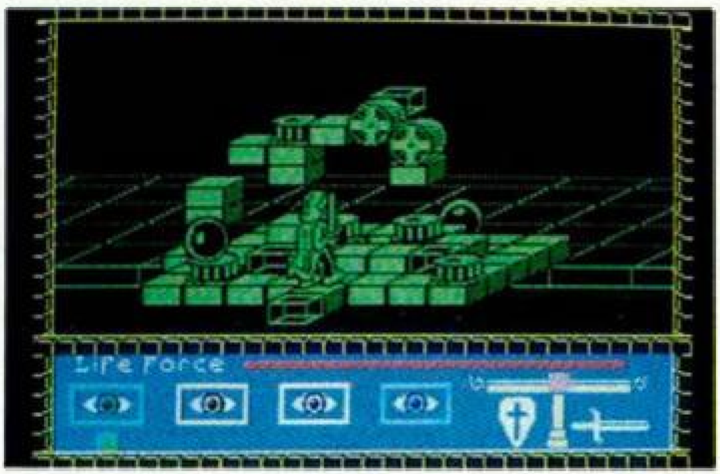
Lo que tenéis que hacer es lo que a continuación se detalla:

— En cualquier momento del juego detenerlo pulsando la tecla de SPACE.

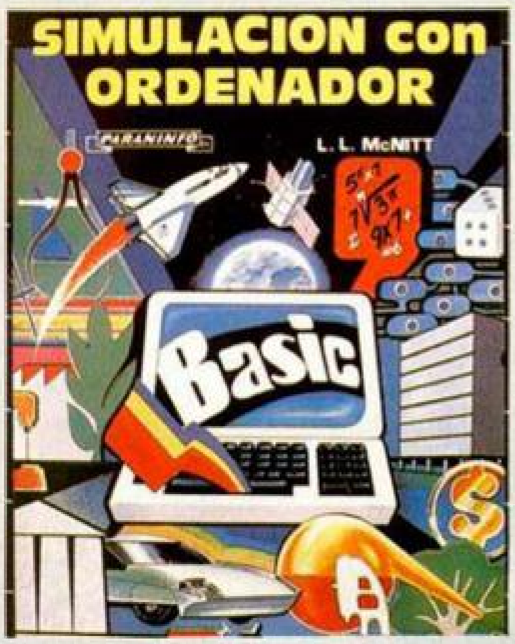
— Pulsad simultáneamente las teclas Capshift y L. Sonará un pitido.

— Cuidado ahora estamos en el momento clave. Pulsad letra a letra la palabra LENIN (primero la L, luego la E, la N, etc...) y obtendréis, ¡Oh maravilla!, vidas infinitas.

¿No está mal, verdad?



LIBROS



SIMULACION CON ORDENADOR

L.L. McNitt
Editorial Paraninfo
200 páginas

La simulación con ordenador, en la que se pueden incluir modelos físicos, matemáticos o informáticos, se está divulgando a una gran velocidad entre las industrias, economía, medicina, ingeniería y tecnología espacial. El propósito de este libro es el acercar a los estudiantes y profesionales a los modelos y métodos de simulación. **«Use su computadora para explorar lo desconocido, resolver lo irresoluble y buscar lo imbuscable.»**

Simulación con Ordenador contiene una considerable cantidad de programas ejemplo, los cuales han sido escritos en Basic y posteriormente comprobados en un Osborne I. De esta forma, al utilizar este lenguaje tan universal, los programas pueden ser traspasados a un gran número de ordenadores diferentes sin necesidad de realizar cambios notables. El libro enseña cómo aplicar fácilmente las técnicas más importantes para la resolución de los problemas que se puedan plantear en los negocios, finanzas, ciencia e incluso en la enseñanza y el divertimento.

Para el uso y aprendizaje de los contenidos del libro no se requiere un conocimiento previo muy extenso ni de los temas relacionados con la programación ni con las matemáticas, por lo que esta guía puede ser usada como manual autodidáctico además de como un texto en simulación con computadora.

LOS JUSTICIEROS DEL SOFTWARE



LAS TRES LUCES DE GLAURUNG

Muchos son ya los programas que han pasado bajo la mirada escudriñadora de nuestros jueces. Estos en su gran mayoría han sido realizados en Gran Bretaña, encontrándonos en muy escasas ocasiones ante un programa nacional. Pero nuestros justicieros no conocen de países ni banderas; sus juicios son siempre imparciales e inapelables.

«Gráficos sin pretensiones»

● POSITIVO

Los saltos son exagerados y difíciles de calcular. Es posible eliminar a los enemigos sin gastar armas. Estos tienen vida propia y no están fijos en una pantalla. Muy original el detalle de los burladeros, así como su movimiento y sonido.

● NEGATIVO

Gráficos sin pretensiones y sonido pobre. La temática es poco original resultando monótono y poco estimulante ya que hay que recorrer muchas pantallas para encontrar algo útil.

Puntuación: 5

Alberto A. Cifrián Miranda
Torrelavega (Cantabria)

«Gran poder de adicción»

● POSITIVO

Tiene gran diversidad de enemigos y estupendos gráficos así como la posibilidad de cambiar de aspecto a nuestro personaje. Gran poder de adicción.

● NEGATIVO

El movimiento del personaje resulta algo lento en las habitaciones donde se encuentran muchos enemigos. En ocasiones los gráficos de los enemigos se mezclan con otros. Las pantallas son muy similares entre sí.

Puntuación: 8,5

Angel Pérez Barreiro
Sabadell (Barcelona)

«Movimiento perfecto»

● POSITIVO

Es un programa de alta calidad. Los gráficos son muy originales y están bien diseñados. El movimiento es perfecto. Tiene detalles que hace que tengas todos los sentidos alerta, por lo que resulta muy adictivo. Para ser el primer programa de Erbe se puede colocar entre los mejores juegos británicos sin ningún problema.

● NEGATIVO

Tal vez la dificultad es baja debido a la escasez de armas, mientras que lo difícil es conseguir nuestro objetivo por el número tan extenso de pantallas.

Puntuación: 9

Jorge Segarra Estébanez
Aguilar de Campóo (Palencia)

«Muy buenos gráficos»

● POSITIVO

Las posibilidades de acción de nuestro personaje son muy amplias y buenas. Los gráficos están muy bien conseguidos al igual que el sonido. Tiene un gran poder de adicción.

● NEGATIVO

Las pantallas son iguales entre sí. Recuerda demasiado a otros juegos y sólo podemos utilizar el teclado para el manejo.

Puntuación: 8,5

Mario Melinat
Marchamalo (Guadalajara)

«Aceptable dificultad»

● POSITIVO

Es un juego bastante original tanto en el planteamiento de éste como en sus gráficos. Consta de una buena mezcla de colores. La melodía de presentación es muy buena, evocándonos a la época medieval. Tiene una aceptable dificultad con el número de vidas que contamos.

● NEGATIVO

El movimiento un poco brusco. El sonido del movimiento del personaje acaba haciéndose pesado y monótono.

Puntuación: 8,5

Luis García Sánchez
Pto. Sta. María (Cádiz)

«Demasiados enemigos»

● POSITIVO

Las pantallas son gráficamente muy graciosas, lo que hace que el juego sea adictivo. Es muy fácil controlar los movimientos del personaje que son demasiado rápidos.

● NEGATIVO

La historia es poco original y las armas de las que disponemos son limitadas para la gran cantidad de enemigos que nos acechan.

Puntuación: 9

Pedro Revilla Ortega
Burgos

«Movimiento muy bueno»

● POSITIVO

Gráficos bastante bonitos, siendo de destacar el tamaño de los sprites. Movimiento muy bueno y una de las mejores cosas, junto con los gráficos, que presenta este programa.

● NEGATIVO

El sonido, es en general bastante pobre. Otro aspecto negativo es la no inclusión de redefinir las teclas. La mecánica del juego es muy adictiva pero poco original y sigue la línea clásica de las videoaventuras.

Puntuación: 8

Misericordia Pagues Gil
Reus (Tarragona)

«Rapidez de respuesta del teclado»

● POSITIVO

La rapidez de respuesta de Redhan, así como la tonadilla que se oye después de cargar el juego. Gráficos muy buenos y simples.

● NEGATIVO

Gran dificultad a la hora de recorrer todas las habitaciones del castillo. Es muy difícil llegar a controlar la técnica del choque.

Puntuación: 9

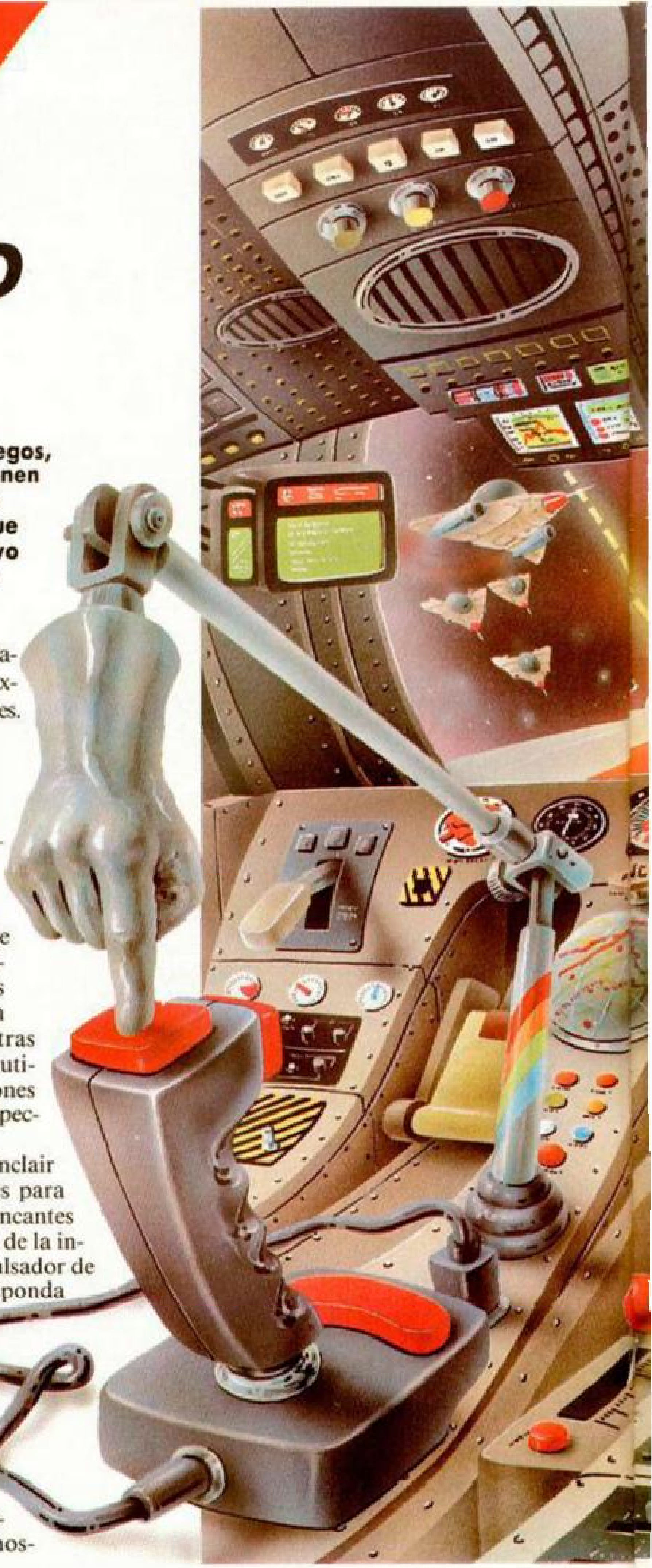
Santiago Colás
Zaragoza



LAS INTERFACES Y EL AUTODISPARO DEL JOYSTICK

Primitivo DE FRANCISCO

Cuando se utiliza el joystick en programas de juegos, los dedos pulgar o índice de la mano derecha tienen una actividad constante en los diparos continuos que se hacen contra los enemigos electrónicos que se mueven inquietos por la pantalla. El dispositivo de autodisparo viene a evitar la fatiga «digital» aumentando al mismo tiempo la puntería.



Debido a que el Spectrum no dispone en su interior de interface de joystick, varias firmas comerciales se han lanzado a fabricar interfaces con criterios diferentes, lo que ha dado lugar a un gran descontrol en su utilización, y por supuesto, a incluir las correspondientes opciones por software al principio del juego.

Todos los fabricantes partieron del tipo de stick o palanca para juegos que ya existía previamente y que comercializó ATARI. El éxito alcanzado posteriormente ha propiciado su estandarización. Este joystick usaba cuatro pulsadores para las cuatro direcciones geográficas más un quinto que hacía de disparo. El conector era de tipo CANON de 9 terminales en el cual la distribución con los pulsadores era: **1: Pulsador hacia arriba, 2: Pulsador hacia abajo, 3: Pulsador izquierdo, 4: Pulsador derecho, 6: Disparo, 8: Común, 7: +5 Voltios, 5: Potenciómetro A y 9: Potenciómetro B.** Estos dos últimos son añadidos para entregar valores analógicos procedentes de potenciómetros de mando.

Existe otro tipo de joystick que utiliza exclusivamente potenciómetros para obtener la información de movimientos. En éstos, la palanca se encuentra ligada mediante un mecanismo a dos potenciómetros colocados físicamente en ángulo recto, siguiendo los ejes X e Y. Este método requiere un convertidor analógico-digital en la interface, presenta una mayor precisión de posicionamiento, pero su manipulación es más lenta para juegos y sobre todo, su mecanismo más delicado frente a los movimientos impetuosos de la palanca que se producen en el fragor de la jugada.

Los joysticks más populares para Spectrum son exclusivamente de pulsadores.

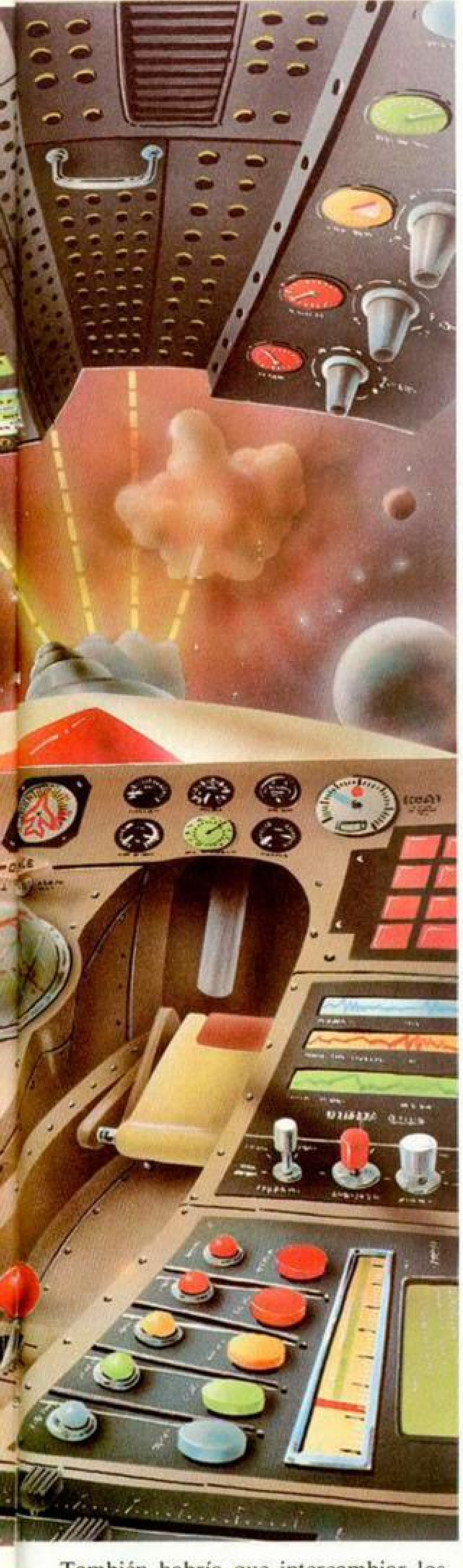
Interfaces para joystick

Resaltamos como más introducidos en el mercado el Kempston, Sinclair o interface 2 y de cursor o PROTEK. Los dos últimos simplemente emulan algunas teclas mientras que el tipo Kempston utiliza una de las direcciones de puerto libres en el Spectrum.

La Interface-2 de Sinclair incluye dos conectores para poder jugar dos contrincantes entre sí. La circuitería de la inrface hace que cada pulsador de la palanca se corresponda con una tecla de la fila superior del teclado.

Al Joystick-1 le corresponden las teclas 1, 2, 3, 4 y 5, al Joystick-2 las teclas 6, 7, 8, 9, y Ø, según la distribución mostrada en la **figura 3.**

En la **figura 1** se muestra el circuito de una de las palancas en el cual un circuito integrado con cinco triestrados hábiles vuelca el contenido de los pulsadores del joystick en el bus de datos. Esto ocurre cuando se produce el direccionamiento de puerto (IORQ) en lectura (RD) con el bit de dirección AØ. Con lo cual se imita el direccionamiento del teclado. Un segundo bit de dirección A11 o A12 habilita el joystick izquierdo o el derecho respectivamente, ya que en el teclado estos dos bits también direccionan la semifila izquierda o derecha de la línea superior del teclado (la fila de los números). Cada semifila se compone de cinco teclas. En la **figura 1** sólo se representa el circuito de uno de los joysticks, el otro sería similar cambiando el bit A11 por A12 de la puerta OR.



También habría que intercambiar los pulsadores con respecto al bus de datos como se aprecia en la **figura 3.** Para más detalles ver la revista número 8 de MICROHOBBY.

Si se elige la opción Sinclair en el menú de joysticks también pueden usarse las teclas de la fila superior al tiempo que la palanca, pudiéndose comprobar en cualquier juego.

La Interface Protek o cursor, hace corresponder los pulsadores de la palanca con las teclas de los cursores del teclado, esto es 5, 6, 7, y 8 en las mismas direcciones indicadas por las flechas inscritas en dichas teclas. El pulsador de disparo se corresponde con la tecla Ø.

Esta interface también imita al teclado, por lo tanto usará las señales citadas anteriormente para el tipo Sinclair (IORQ, RD y AØ). En la **figura 3** se muestra la correspondencia citada. Obsérvese que todos los pulsadores están asignados a la semifila derecha excepto el de movimiento a la izquierda. Esto se traduce en que para la total decodificación de los pulsadores de la semifila derecha se emplea el bit A12 y para el pulsador de movimiento a la izquierda se necesita el bit A11 por pertenecer a la semifila izquierda.

El esquema eléctrico de este tipo de interfaces se muestra en la **figura 1.** Los bits A11 y A12 discriminan los pulsadores asignados a cada semifila. Según lo dicho anteriormente, si se elige la opción Protek, funcionarán simultáneamente el joystick y las teclas de los cursores más la del cero.

La interface Kempston no imita al teclado, sino que mediante un puerto diferente habilita sus triestrados para leer el estado de los pulsadores (**figura2**).

La interface Kempston utiliza el bit A5 como puerto. Cuando este bit se hace cero, junto con las señales IORQ y RD, el joystick es direccionado y leído.

A diferencia de las interfaces que imitan al teclado, la interface Kempston interpreta que se ha movido la palanca en una cierta dirección cuando su bit correspondiente en el bus de datos es un 1. Por ejemplo, si se ha llevado la palanca hacia la derecha, en el bit DØ aparecerá un 1 cuando se lee el puerto DFh o 223 en decimal. Esta es precisamente la dirección real de la interface Kempston: DFh es el resultado en hexadecimal de poner los bits AØ, A1, A2, A3, A4, A6 y A7 a uno y únicamente A5 a Ø. Existen naturalmente otras muchas direcciones que poseen el bit A5 a Ø, direcciones que también habilitará la interface, pero al ser usadas éstas por otros periféricos junto al Kempston, se producirán errores de interpretación en la lectura del puerto.

La **figura 2** muestra dos formas de obtener la interface Kempston, en función del tipo de triestrados empleados. Cuando éstos no son inversores (por ejemplo el 74LS244) el común de los pulsadores habrá de ir a +5V para que el bit leído procedente de un cierto pulsador sea efectivamente un uno. Por el contrario si emplea el circuito integrado 74LS240 que contiene 8 triestrados

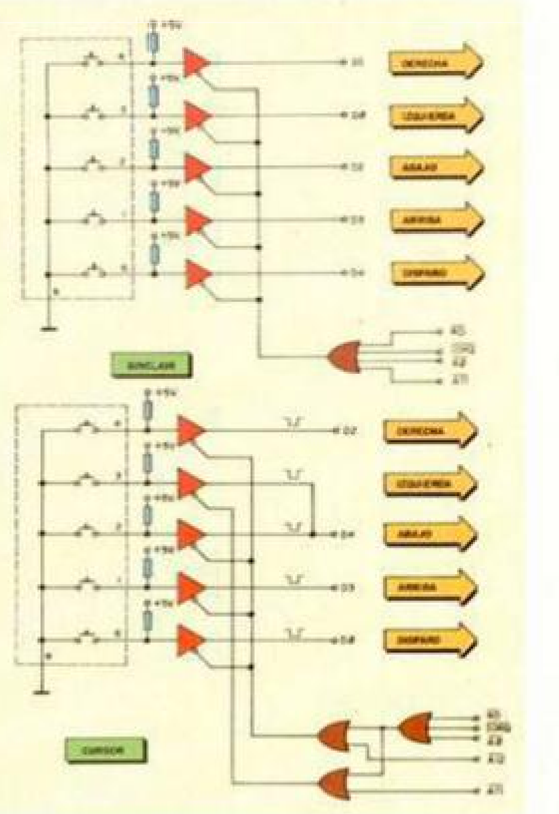


Figura 1.

inversores, el común de los pulsadores del joystick habrá de estar conectado a tierra. La **figura 2** muestra todos los detalles y filosofía de conexión de este tipo de interface.

El autodisparo

En el cuerpo de algunos joysticks existe una opción extra de autodisparo o disparo repetitivo. Esta opción se selecciona llevando un conmutador a la posición AUTO o se inhabilita llevándola a MANUAL.

El disparo repetitivo se obtiene gracias a un oscilador que existe en el circuito impreso de la propia palanca. La frecuencia de este oscilador es aproximadamente de 80 Hz. Sus impulsos irán hacia el terminal de disparo del conector del joystick (Pin 6) siempre que se



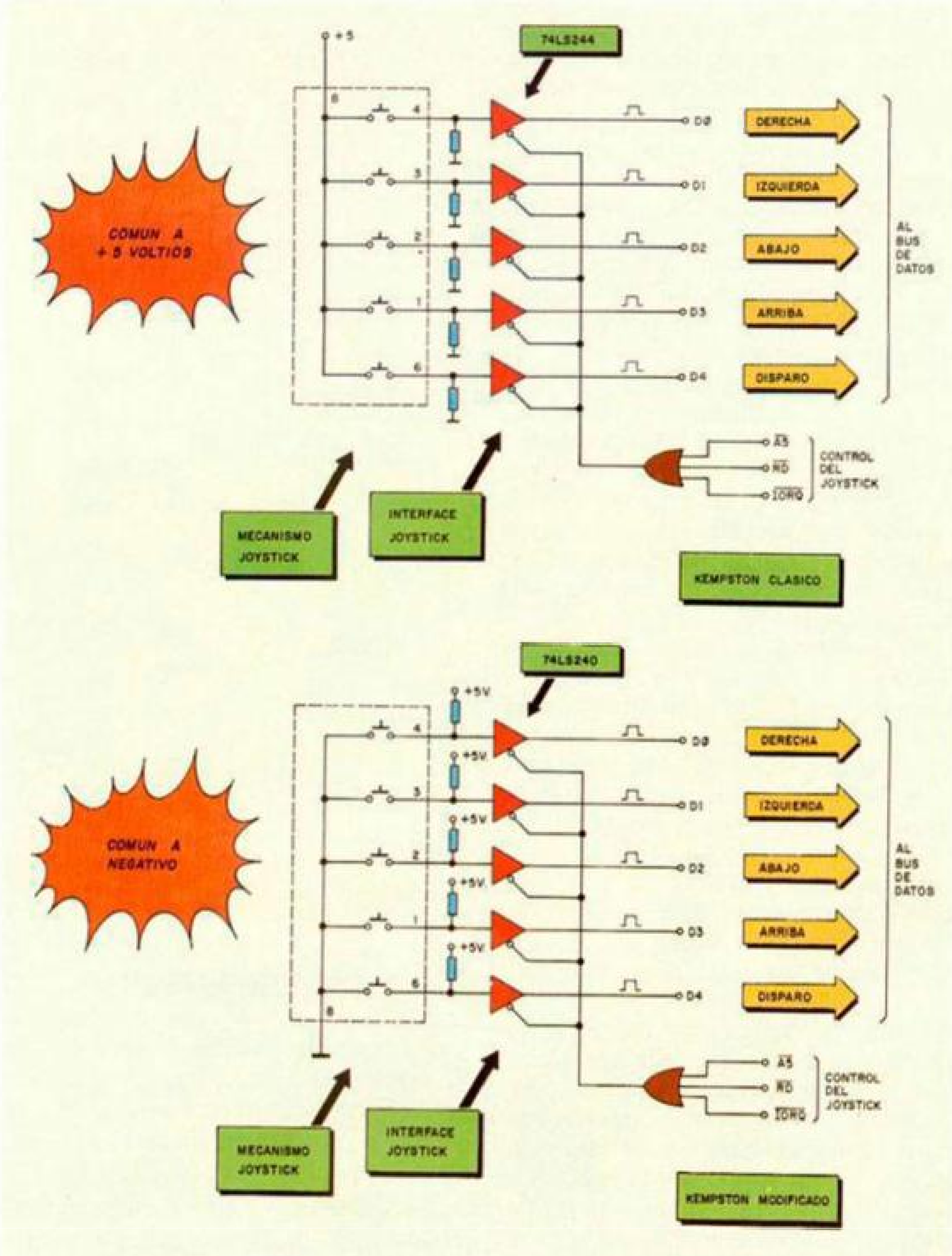


Figura 2.

haya seleccionado el modo AUTO.

El oscilador puede estar hecho con el famoso circuito integrado multivibrador 555 y unos cuantos componentes más, tal como se muestra en la **figura 4.** El hecho de incluir el oscilador en la propia base del joystick es para que éste sirva con cualquiera de las interfaces mencionadas; pero ahí viene el problema, porque la interface ha de cumplir NECESARIAMENTE dos condiciones: Entregar +5V por el terminal 7 del conector del joystick y que el común de los pulsadores en la interface vaya a tierra o negativo, porque de estos dos terminales obtiene el oscilador 555 la alimentación. La mayoría de las interfaces no entregan +5V por el terminal citado; por otra parte, en algunas interfaces Kempston el común de los pulsadores no va a tierra, sino a +5V como hemos dicho anteriormente. Por tanto, si se adquiere un joystick con disparo automático, lo más probable es que no funcione en nuestra interface.

Joystick sin disparo automático

Si nuestro interface es Kempston y tiene su común (Pin 8) a +5V no se podrá hacer ninguna modificación. Esto se puede comprobar abriendo la caja y observando con un polímetro si este terminal efectivamente es positivo.

Pero si, por el contrario, nuestra interface tiene el común a negativo, sólo habrá que soldar un hilo desde el terminal 7 del conector canon hasta la pa-



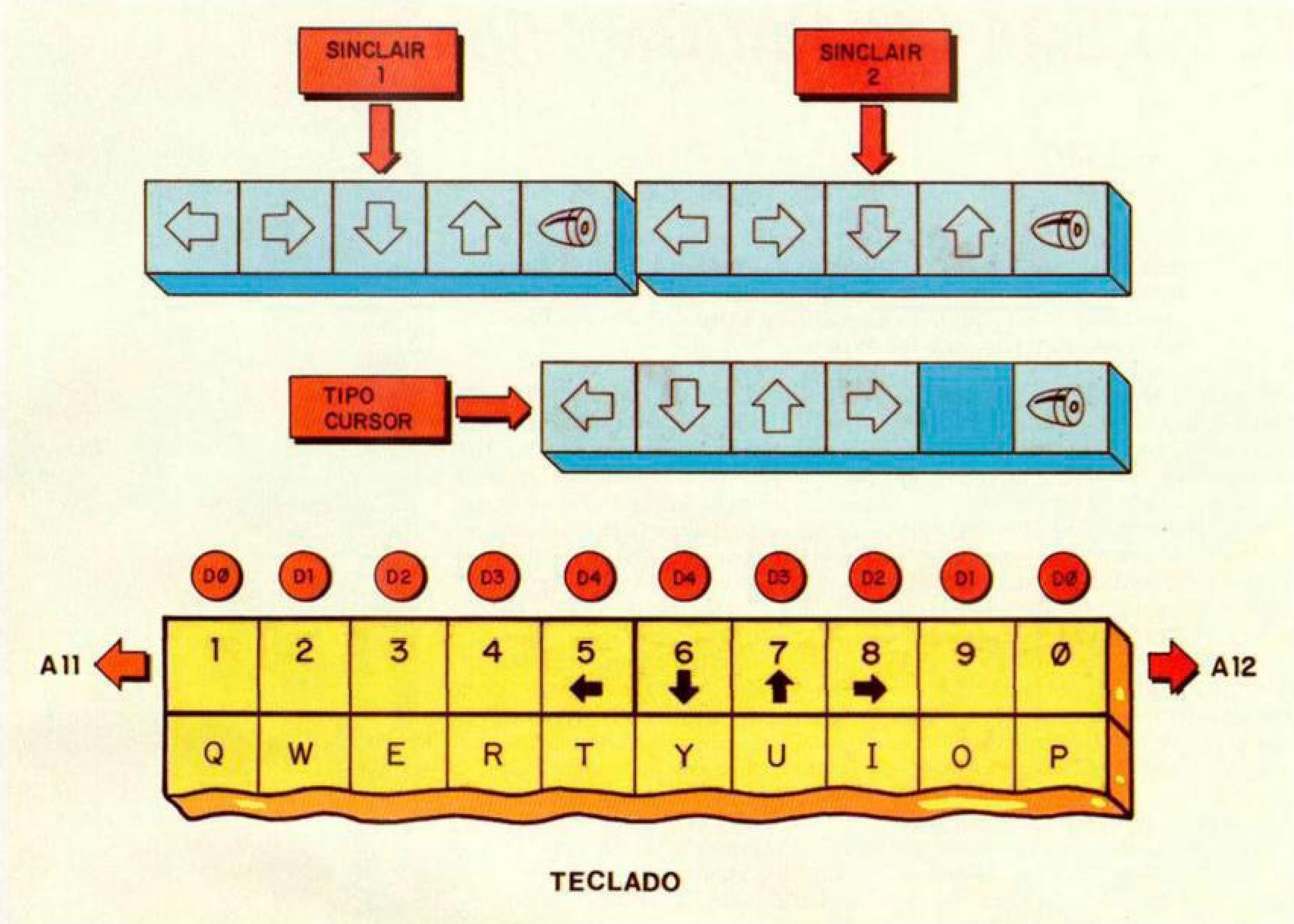


Figura 3. Cada uno de los pulsadores del joystick se corresponde con una tecla de la primera fila del teclado mediante la Interface-2 de Sinclair o la Interface de tipo cursor. Los círculos expresan el bit del bus de datos por donde son leídos.

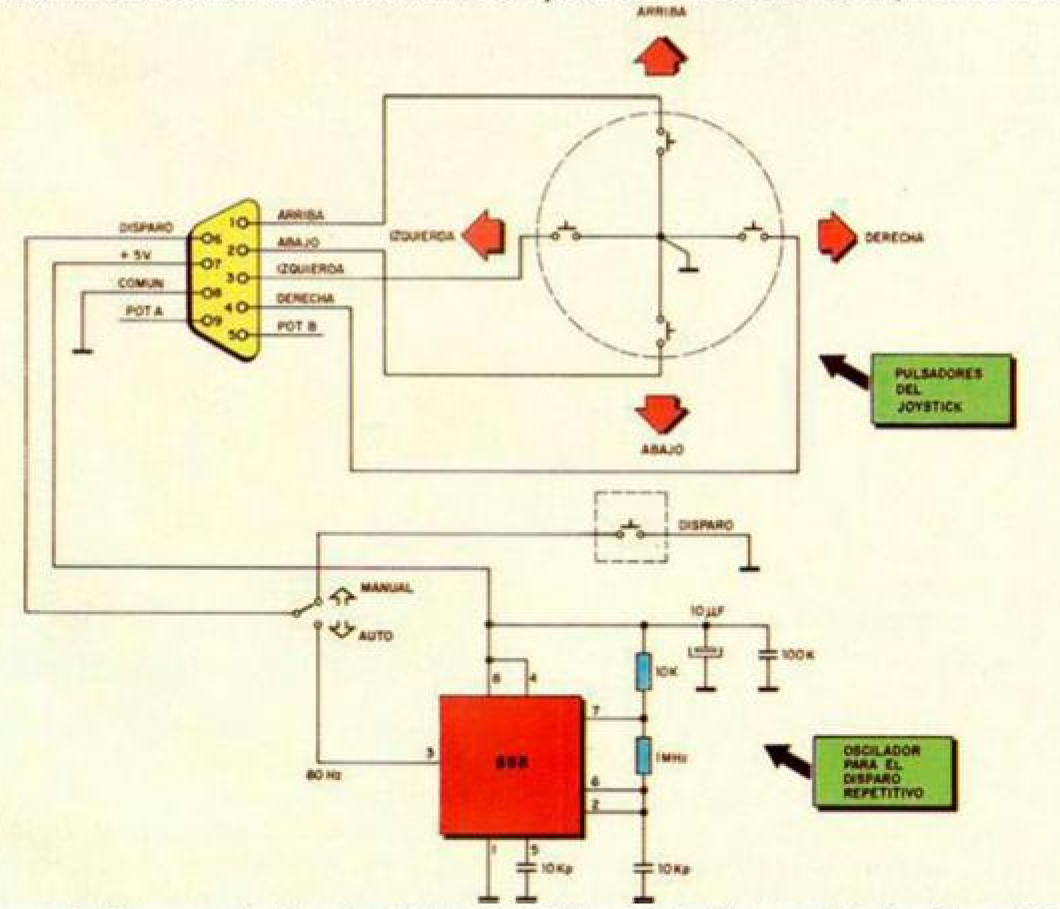


Figura 4. Esquema de los joysticks que utilizan autodisparo. El circuito está incluido en la base de la palanca. Para que el autodisparo sea efectivo la interface ha de entregar tierra por el terminal 8 y +5V por el 7 del conector *CANON*. Mediante un conmutador se pasa a disparo manual o disparo repetitivo.

ta 14 del 74LS32 que de seguro existe en el circuito, para obtener autodisparo, además incluir el 555 en el interior de la palanca tal como se muestra en la **figura 4.**

El oscilador podrá ser soldado aéreamente en cualquier lugar de la placa de circuito impreso. El conmutador puede ser uno miniatura de palanca que se incluirá en la caja de plástico con tan sólo efectuar la adecuada perforación. En posición «auto», producirá constantemente disparos asegurando la certeza del impacto.

Una acción equivalente consistiría en disparar en ráfaga, presionando y soltando el botón de fuego continuamente. Es decir, no es lo mismo que dejar el botón pulsado, truco que no funciona en la mayoría de los juegos y por otra parte, para llevar a cabo esta acción no haría falta tanto lío. Esperamos que con este montaje vuestros dedos queden de una vez por todas libres de tan pesado e inútil ejercicio.

EL SISTEMA "FILMATION" (III)

José Manuel LAZO

El método «Filmation» de programación es considerado como el más sofisticado de los conseguidos hasta ahora. Dentro de este sistema, una de las claves más importantes es el control de los «sprites». En este capítulo explicamos exhaustivamente todo lo referente al manejo de los mismos.



Precisamente uno de los factores a tener en cuenta en la estructura del sprite es la relación entre sus partes «**opacas**» y «transparentes», es decir, las que permiten ver a su través y las que no.

Si os fijáis en la **figura 2,** podréis comprender la manera de determinar las partes opacas del sprite y ponerlas adecuadamente en la pantalla sin perjudicar el resto: en la máscara se halla marcada una zona opaca con un 1, y una transparente con un Ø. Esta condición se invierte con una operación CPL de forma que lo que en la pantalla ha de permanecer invariable está representado en el byte que nos toque procesar de la máscara con un 1.

Si cogemos el byte de pantalla y hacemos una operación **AND** tendremos que en este byte habrá permanecido invariable la parte del mismo que corresponda a una zona transparente del sprite y la opaca que se habría puesto a Ø. Ya sólo nos queda realizar un **OR** con el sprite propiamente dicho y volver a poner el byte producto de todas estas operaciones otra vez en la posición que le correspondía en la pantalla. Así de sencilla es la fórmula del sprite:

SPRITE = (CPL MASCARA) AND PANTALLA OR GRAFICO

Esto se puede apreciar en la línea 184Ø del listado 3 (ver número anterior): en A cogemos el byte de máscara, luego lo invertimos con un CPL y lo «ANDeamos» con la dirección de pantalla, en la línea 187Ø; lo guardamos en la pantalla como almacenamiento temporal y cogemos en el registro A el byte del gráfico. Por último, en la línea 191Ø lo «ORreamos» con el byte de pantalla y situamos el resultado en la misma.

Alta resolución en vertical

Lógicamente, unas rutinas que se precien de gestión de sprites han de trabajar en la máxima resolución del ordenador, esto es, pixel a pixel. La alta resolución en sentido vertical es más sencilla de conseguir que en sentido horizontal.

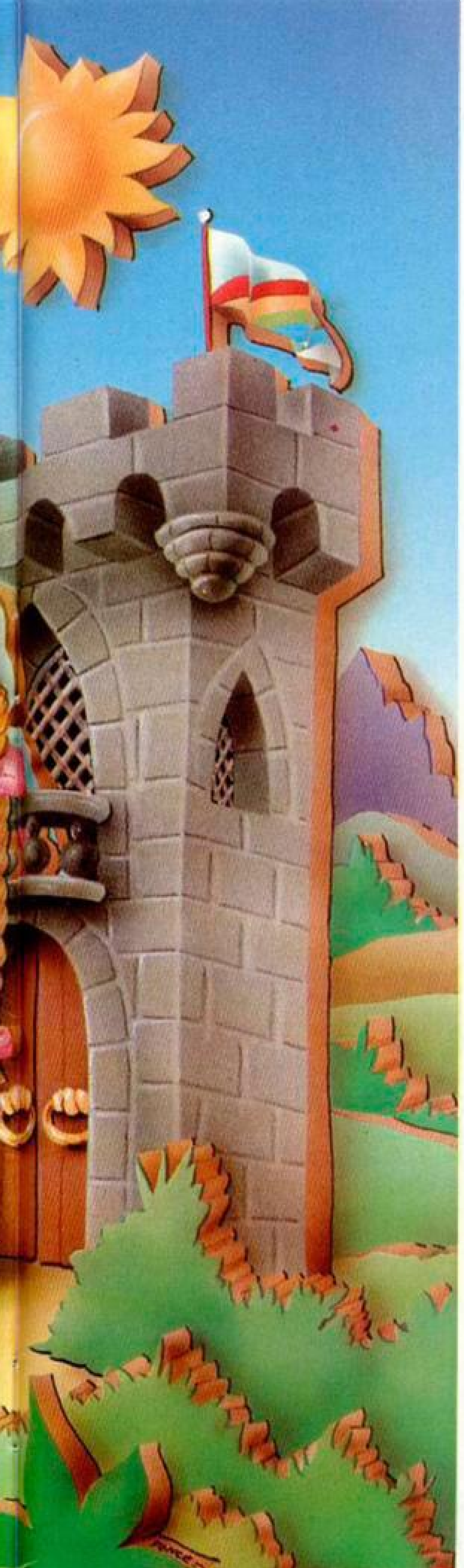
Encontrar la dirección de la memoria de pantalla correspondiente a una determinada posición **Y** es cosa fácil ya que lo realiza una rutina de la ROM: **pixel-adress.** No vamos a comentar ahora cómo funciona debido a que ya se explicó en el artículo **Rutinas Gráficas,** escrito por el mismo autor en el MICROHOBBY número 83 y sucesivos. Podéis apreciar cómo dicha rutina es llamada en la línea 236Ø y 23Ø del listado 3.

El problema reside en calcular la dirección del scan que se halle debajo de uno determinado. Debido a la organización de pantalla del Spectrum este cálculo no se puede realizar con una simple suma, sino que ha sido necesario desarrollar una rutina que, sin perjudicar ningún registro calcule, para un valor de pantalla contenido en HL el que se halle debajo. La rutina que hace esto es la que se encuentra en la línea 376Ø y sucesivas del **listado 3.**

Pueden existir tres condiciones determinadas para las cuales hay que realizar sendos cálculos distintos:

— Por una parte que el próximo scan se encuentre en la misma posición de carácter que el primitivo: después de estudiar la organización de pantalla nos damos cuenta que esta condición se puede determinar por los tres bits de menor peso del registro de mayor peso que esté apuntando a la pantalla, en este caso HL. Si estos tres bits se encuentran elevados es que el scan primitivo es el último de un carácter y el siguiente está en otra posición de carácter. La manera de determinar esto, línea 378Ø, es enmascarar estos tres bits con una operación AND y comparar con 7. Para calcular la próxima posición de pantalla si el scan está en el mismo carácter sólo hay que aumentar en una unidad del registro de mayor peso que apunta a la pantalla.

— En segundo lugar se puede dar el caso de que el próximo scan se halle en otro carácter. Lo primero que tenemos que hacer ahora es verificar si, además de cambiar de carácter, hemos cambiado de tercio de pantalla. Esto se detecta porque el registro de menor peso contiene un valor



superior a EØh *(línea 385Ø)*. El cálculo para cambiar de carácter es también muy sencillo: sumar al registro de menor peso 32 y restar al de mayor 7 *(líneas 387Ø y sucesivas)*.

— Por último, en el caso de que cambiemos de tercio y carácter hay que hacer un tercer cálculo muchísimo más sencillo: sumarle al registro HL el valor 32.

Así de sencilla es la rutina para calcular la próxima dirección de un scan. Si os dais cuenta al principio de la rutina se salva el registro A y los flags en la pila con la cual conseguimos que esta rutina realice todos sus cálculos sin modificar en nada ningún registro, salvo el HL que es, lógicamente, el que nos interesa modificar.

Alta resolución en horizontal

Este problema es bastante más complejo de resolver como ahora veremos. En el listado 3 hay una rutina, llamada **SCAN**, que es común para todas y que tiene como misión el «spritear» un scan del gráfico que queramos imprimir en alta resolución en horizontal. En esta rutina podremos apreciar la técnica utilizada.

En primer lugar fijaros en la **figura 3**. Representa las operaciones que hay que hacer con un byte del gráfico para ponerlo en el sitio adecuado de la pantalla. La alta resolución en horizontal tiene como problema principal el que un byte del sprite va, en la mayoría de los casos, entre dos bytes de la memoria de pantalla, es decir, se encuentra «partido» en dos. Además hay que saber por dónde hay que partirlo y desarrollar una rutina muy, muy rápida que lo haga. Piénsese que este es el escalón más inferior de la rutina y que si no es rápido, el efecto de lentitud se multiplicará por un factor bastante elevado.

Los dos rectángulos de la parte superior representan dos bytes de la memoria de pantalla, por ejemplo 4ØØØ h y 4ØØ1h y los dos de la parte inferior, uno sólo del gráfico que pretendemos imprimir, el cual va a ir situado en la coordenada 3 en X. Como se ve el byte del gráfico va entre estos dos de la pantalla. Si lo pusiéramos sobre el de la izquierda nos damos cuenta que habría que desplazar el *gráfico* 3 bits a la derecha. Y si lo ponemos sobre el de la derecha habría que desplazarlo 5 bits a la izquierda.

De esto se deduce el siguiente algoritmo: hay que hacer dos operaciones con cada byte del gráfico, en la primera, con el primer byte de pantalla hay que desplazar a la derecha el byte del sprite tantas veces como marquen los tres bits de menor peso de la coordenada X donde deseamos imprimir. En la segunda operación, que se realiza sobre el próximo byte de la pantalla, hay que hacer un desplazamiento del mismo byte del gráfico tantas veces como el resultado de 8-los tres bits de menor peso de la coordenada X.

Este último byte de pantalla es el primero para el próximo byte del gráfico debido a que ha quedado un espacio a la derecha sin imprimir.

En realidad es un poco más complicado; piénsese que no sólo hay que hacer esto, sino que, además, hay que takear la pantalla coherentemente y aplicar la fórmula del sprite en las dos operaciones, así como hacer también las rotaciones sobre la máscara.

Las dos rutinas de rotaciones, situadas en las líneas 135Ø y 147Ø, se han puesto por separado para optimizar en lo posible el funcionamiento de las mismas a costa de un consumo de memoria. Asimismo, estas rutinas no perjudican ningún registro, nada más que el A, que es el que se pretende rotar y no requiere ningún valor de entrada.

El buffer del gráfico en donde se guarda lo takeado de pantalla, es en este caso, un carácter más grande que el gráfico mismo, esto es así porque al estar el gráfico entre dos bytes siempre sobra un trozo de byte por una parte que, junto con el otro trozo por la otra forma un byte de pantalla que hay que guardar porque el sprite lo corrompe. El takeo se realiza entre una rotación y la siguiente, salvo al principio del volcado del scan. Esto lo indica la variable SWIC que al llamar a la rutina SCAN ha de estar a 1.

Esto es básicamente todo lo que se puede contar de la rutina de volcado de sprites, para usarla sólo hay que cargar el registro BC con las coordenadas de la pantalla que coincidan con la esquina superior izquierda del sprite que queremos imprimir, el registro HL con la dirección donde se encuentre la tabla del gráfico y hacer un CALL SPRITE; entonces veremos cómo nuestro gráfico es puesto en pantalla de una forma fabulosamente rápida.

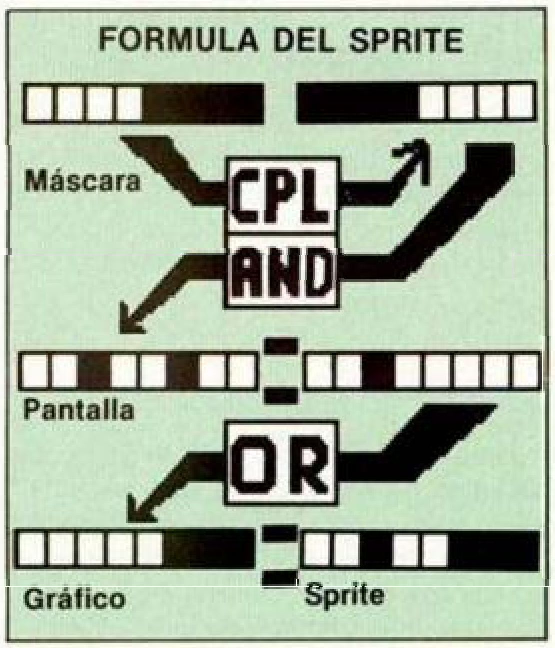


Figura 2

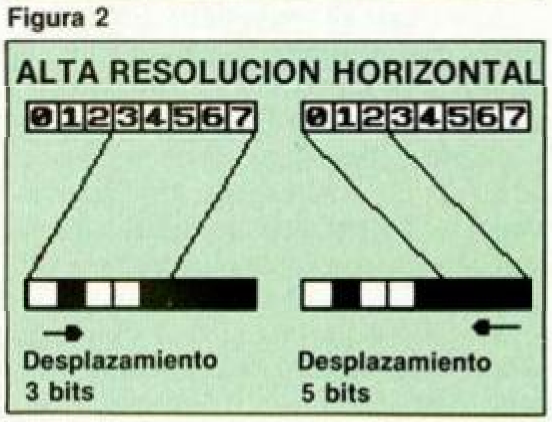


Figura 3

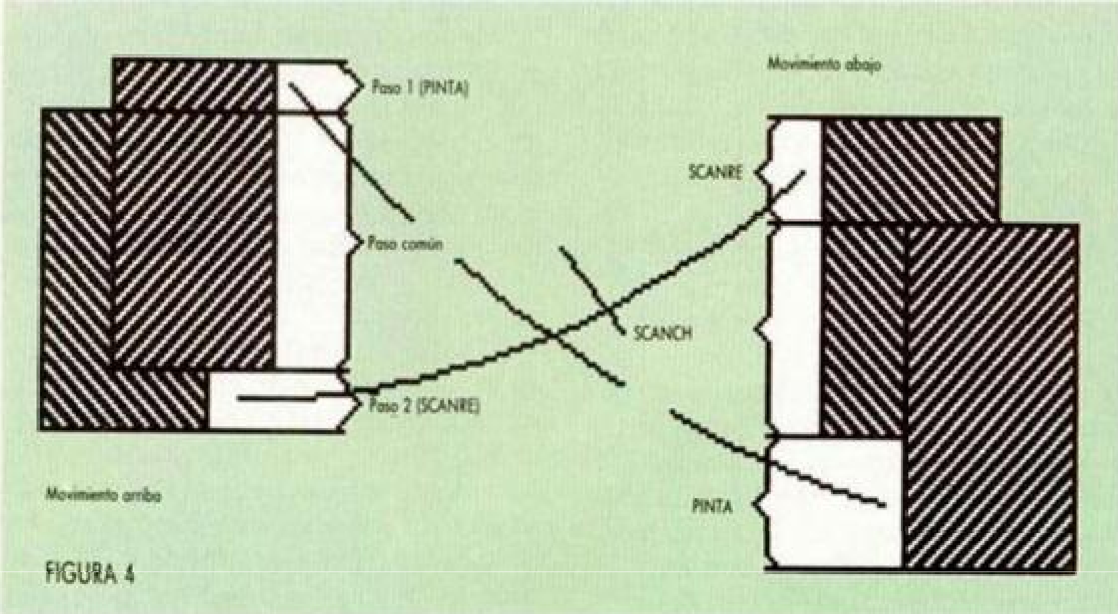


FIGURA 4

La rutina borradora

Esta rutina, situada en la línea 990 y sucesivas, se encarga de borrar un sprite cualquiera que tuviésemos en la pantalla; es la más sencilla de usar y entender. Sólo tenemos que cargar HL con la dirección donde se encuentre la tabla del gráfico y llamarla. Lo que hace es volcar scan por scan el contenido del buffer que, previamente, hemos cargado con la rutina SPRITE en la pantalla. Realmente tiene pocas complicaciones y una vez entendidas las explicaciones anteriores sobre el funcionamiento de la rutina volcadora no debe haber ningún inconveniente con ésta.

Aquí no es necesario hacer cálculos para alta resolución en horizontal, ya que se vuelca por caracteres. La rutina que utiliza es NEXTHL que, como ya explicamos, se encarga para un determinado scan, de calcular la dirección del que se encuentre inmediatamente debajo.

Rutina de movimiento

Hemos dejado para el final las explicaciones de la filosofía de funcionamiento de la rutina de movimiento de un sprite en la pantalla debido a que es la más compleja y que realiza las operaciones más inteligentemente.

Probamos sistemas distintos para mover el sprite por la pantalla, unos adolecían de parpadeo y otros de lentitud. Tras múltiples devaneos, llegamos a la conclusión de que la única manera de mover el sprite por pantalla rápidamente y sin parpadeos era borrar el que estuviera en la antigua posición a la vez que imprimimos el de la nueva.

Si os fijáis en la **figura 4** os daréis cuenta rápidamente de la filosofía que hemos seguido para su desarrollo. El tema era que si, por ejemplo, queremos mover nuestro muñeco un par de pixels hacia abajo restableciéramos los tres primeros scanes del gráfico antiguo, luego empezamos a takear las posiciones de pantalla del gráfico moderno a la vez que lo vamos imprimiendo y restableciendo el antiguo. Por último sólo volcamos los dos últimos scanes del gráfico moderno.

Lástima que este algoritmo que funciona tan bien si desplazamos el sprite hacia abajo se muestra incompetente, si lo movemos hacia arriba o hacia uno de los lados por lo que ha sido necesario detectar a la entrada de la rutina MOVE, líneas 2690 y sucesivas, si el movimiento es hacia arriba, hacia abajo o sin movimiento vertical ya que para las tres condiciones es preciso usar un algoritmo distinto. Si nos centramos en la parte izquierda de la **figura 4** veremos que está representado el caso de que el sprite se mueva hacia arriba (subrutina situada a partir de la línea 2970).

Primero tendremos que volcar una serie de scanes del sprite en la nueva posición; este número de scanes viene determinado por la resta de la nueva coordenada **Y** con respecto a la antigua, lo cual se puede ver en el bucle situado a partir de la línea 3040 el cual llama a la rutina PINTA.

Esta rutina (línea 4050) se encarga de pintar un scan del nuevo sprite en la nueva posición sirviéndose de la rutina SCAN. Además va almacenando el contenido de la pantalla en el buffer 1.

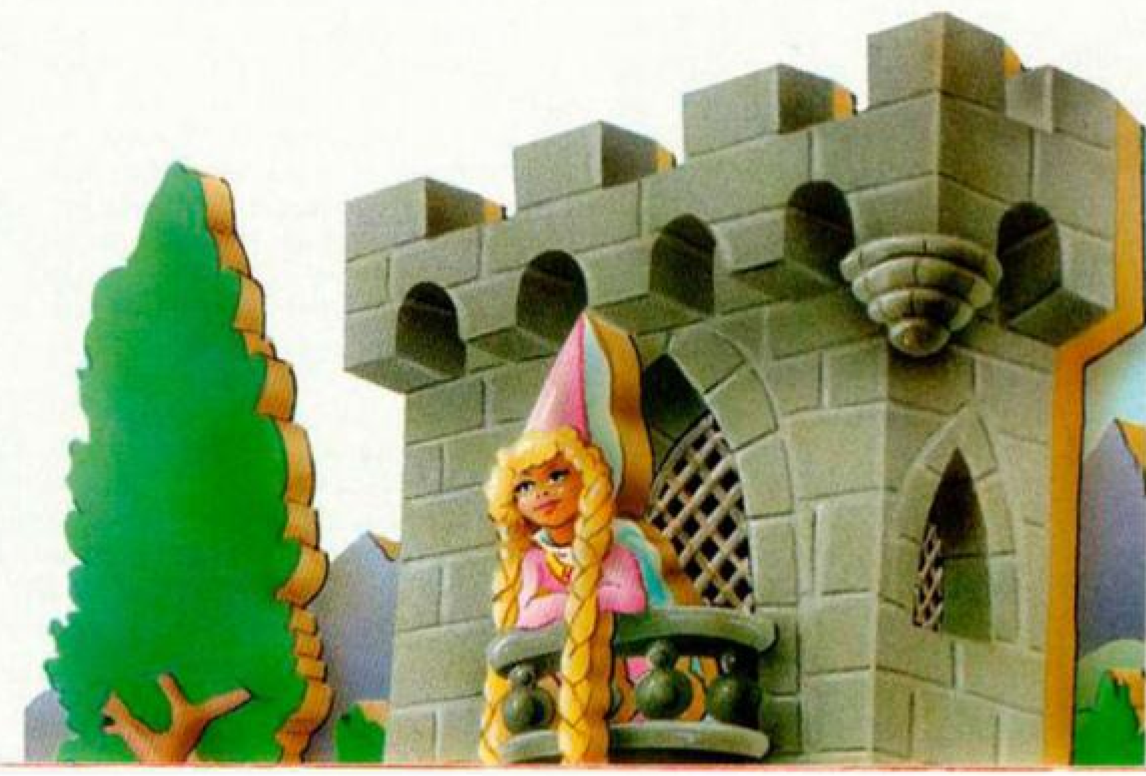
Es de destacar en este momento que cada gráfico, si vamos a usar la rutina MOVE precisa de dos buffers de idéntica longitud. Esto es así para poder ir guardando en el primero y restableciendo a partir del segundo. Lógicamente, la rutina MOVE se encarga de volcar, con la subrutina CHANGE, el contenido del buffer 1 en el 2.

Cuando hayamos terminado con el primer bucle tenemos volcado en pantalla el gráfico en la nueva posición hasta el scan de comienzo del sprite de la vieja posición, en este momento tenemos que empezar a realizar una operación un poco más compleja, línea 3120 y sucesivas, que es restablecer un scan del antiguo gráfico, takear el contenido del scan correspondiente al nuevo sprite, y volcar este scan del nuevo sprite en su posición en pantalla.

El bucle se repetirá tantas veces como scanes haya interseccionados entre el viejo y el nuevo sprite.

El último bucle para este algoritmo está a partir de la línea 3170: sólo tenemos que restablecer los scanes que queden por debajo del sprite en la antigua posición.

Para el caso de que queramos mover el muñeco hacia abajo en vez de hacia arriba, ver **figura 4**, los pasos que hay que seguir son los contrarios, es decir: primero restablecer el trozo no solapado del antiguo sprite por el moderno, luego *«intercambiar»* los scanes del antiguo sprite por el nuevo, y por último, pintar lo que quede del nuevo sprite.



TRUCOS

JOYSTICKS MAS SENSIBLES



Este preciado instrumento casi siempre necesita funcionar a duras penas efectuando arriesgadas maniobras acrobáticas, por lo cual precisa de una sensibilidad que no siempre posee; la solución es muy fácil, basta con seguir estos pasos:

— Se desmonta el joystick, con mucho cuidado de no romper o dañar los cables que unen la palanca al soporte (cables del botón de fuego).

— Entonces si miramos en su interior (en el caso del modelo Quit Soft II), veremos unas lengüetas de metal que al unirse efectúan el contacto necesario para el movimiento.

— Pues bien, si fundimos un poco de estaño sobre la superficie de contacto de la lengüeta inferior, en el circuito impreso, y con sumo cuidado lo extendemos en todos los contactos, de tal forma que subamos la altura de éstos un milímetro, obtendremos una vez montado el joystick un mando de SUPER SENSIBILIDAD.

— Por último, hay que tener en cuenta que el aumento del volumen debe ser proporcional en todos los contactos.

— Una vez terminada la operación limpiaremos las superficies de contacto con un bastoncillo humedecido en alcohol.

José Luis Tajada

INPUT NUMERICO EN CUALQUIER POSICION

En alguna otra ocasión hemos publicado alguna rutina para realizar un Input, pero nunca tan sencillita y fácil de entender. Desde Buenos Aires, en Argentina, nos llega un pequeño truco que nos envía Humberto M.

Esta pequeña rutina en Basic realiza con muy pocas instrucciones un Input numérico o alfanumérico en cualquier lugar de la pantalla.

Su funcionamiento es muy sencillo. Toma cualquier carácter, por lo que aplicando luego la función VAL sobre B$ se obtiene el número deseado siempre que los caracteres que componen el mismo sean procesables por dicha función, por ejemplo: 3 + 4, 17589, 4/3, 5*(4 + 10), etc.

Para caracteres alfanuméricos no hay problema.

INPUT NUMERICO

```
   10 LET b$=""
   20 PAUSE 0
   30 IF INKEY$=CHR$ 13 THEN RETU
RN
   40 LET b$=b$+CHR$ (CODE INKEY$
)
   50 PRINT AT 10,10;b$: GO TO 20
```

CARACTERES INVERTIDOS

José Eugenio Berna Huerta de Zaragoza nos envía una corta rutina en código máquina con la que conseguiremos poner boca abajo fácilmente el juego completo de caracteres.

Para utilizarla basta con teclear el programa cargador en Basic que os ofrecemos, y para que aprendáis cómo se ha realizado la rutina, también damos el listado en lenguaje ensamblador.

LISTADO ENSAMBLADOR

```
10          ORG  64000
20          ENT  $
30          LD   (IY-3),252
40          LD   HL,15616
50          LD   DE,64775
60          LD   B,96
70 CERO     PUSH BC
80          LD   B,8
90 UNO      LD   A,(HL)
100         LD   (DE),A
110         INC  HL
120         DEC  DE
130         DJNZ UNO
140         LD   B,16
150 INC     INC  DE
160         DJNZ INC
170         POP  BC
180         DJNZ CERO
190         RET
```

```
   10 FOR A=64000 TO 64029: READ
B: POKE A,B: NEXT A
   20 DATA 253,54,253,252,33,0,61
,17,7,253,6,96,197,6,8,126,18,35
,27,16,250,6,16,19,16,253,193,16
,239,201
   30 RANDOMIZE USR 64000
```

En este espacio también tienen cabida los trucos que nuestros lectores quieran proponer.

Para ello, no tienen más que enviarlos por correo a MICROHOBBY, Ctra. de Irún km 12,400 28049 Madrid.

CONSULTORIO

«PLOT» y «POINT» en C/M

Me gustaría saber por qué no funciona la siguiente rutina en código máquina, con la que pretendo dibujar un punto en la posición de pantalla 88,88 (PLOT 88,88) si en la 0,0 hay un punto:

```
05        ORG    60000
10        LD     B,88
20        LD     C,88
30        CALL   8910
40        POP    BC
50        LD     A,C
60        CP     1
70        JR     NZ,VOLV
80        LD     B,0
90        LD     C,0
100       CALL   8933
110 VOLV  RET
```

Si hago «PRINT USR 60000» me devuelve «0» si en 88,88 no hay punto y «1» si lo hay, por lo que sé que funciona bien hasta la línea 40, pero no hace nada más.

Andrés JOBACHO - Cádiz

☐La rutina que nos envía está bien planteada pero, suponiendo que funcionara, haría exactamente lo contrario a lo que usted indica, es decir, PLOTearía un punto «0,0» si lo hubiera en 88,88. El problema está en la instrucción «POP BC» que desorganiza la pila de máquina; lo increíble es que, a pesar de ello, consiga retornar. Si utilizara esta rutina varias veces dentro de un programa, lo más probable es que llegara a vaciar la pila y el microprocesador se perdiera al tomar parte de lo que hubiera encima de RAMTOP como dirección de retorno. La subrutina «POINT» devuelve su resultado en la pila del calculador, no en la de máquina, por lo que este resultado no puede ser leído con un «POP». Por fortuna, también devuelve el resultado en «A», y podemos usarlo sin tener que leerlo del stack del calculador. La siguiente rutina hace lo que usted pide:

```
10        ORG    60000
20        LD     BC,0
30        CALL   POINT
40        AND    A
50        RET    Z
60        LD     BC,#5858
70        JP     PLOT
80 POINT  EQU    8910
90 PLOT   EQU    8933
```

El retorno se produce, directamente, desde la rutina «PLOT», no es necesario hacer «CALL» a una subrutina y poner después un «RET», en lugar de eso se puede hacer un «JP» a la rutina y el retorno se produce desde ella, de esta forma, se ahorra un byte.

Trigonometría en código máquina

Me gustaría saber cómo es posible obtener el SIN, COS y TAN desde el código máquina de un número almacenado en el registro «A» de la forma más sencilla posible.

José L. REDREJO - Mérida

☐Las funciones trigonométricas no son lo más adecuado para hacer en código máquina, ya que requieren el uso del calculador del la ROM, lo cual no es, en absoluto, sencillo. El problema más grave no es hallar el SIN o el COS, sino leer el resultado, ya que éste será un número fraccionario entre «0» y «1», y este tipo de números son extremadamente difíciles de manejar en C/M.

No obstante, vamos a indicarle de forma resumida, cómo proceder, aunque le recomendamos un detenido estudio del último capítulo de nuestro Curso de Código Máquina (MICROHOBBY números 92 al 95) donde se tratan las subrutinas de la ROM y el manejo del calculador.

Lo primero que hay que hacer es meter el dato de «A» en la pila del calculador. Esto se consigue llamando a la subrutina "STACK_A" (2D28h). A continuación llamar al calculador (con RST #28) con el literal 1Fh para SIN, 20h para COS y 21h para TAN (para salir del calculador podrá usar el literal 38h). Con esto, tendrá el resultado, en coma flotante, en lo alto de la pila del calculador. El problema es leerlo desde ahí. Para ello, lo más adecuado tal vez sea multiplicarlo por 100, si es un seno o un coseno, y hallarle la parte entera, con lo que quedará un valor entre «0» y «100» que sí puede ser leído en un registro y que será directamente proporcional al seno del número. A continuación, y como ejemplo, le mostramos una rutina que halla el seno del número contenido en «A» y devuelve el resultado también en «A» como el seno multiplicado por 100. Recuerde que el valor de entrada ha de estar en radianes.

```
LD     BC,100
CALL   STA_BC;Guarda 100
CALL   STA_A ;Guarda «A»
RST    #28    ;Llama calcul.
DEFB   #1F    ;Halla SIN
DEFB   #04    SIN * 100
DEFB   #27    ;Halla INT
DEFB   #38    ;Sale de calc.
CALL   FPTOA  ;Lee result.
RET
STA_BCEQU #2D2B
STA_A EQU #2D28
FPTOA EQU #2DD5
```

La rutina queda muy bonita, pero es absolutamente inútil (salvo como ejemplo) ya que el dato de entrada tiene que estar en radianes, por lo que la precisión del cáculo dejaría mucho que desear. Lo cierto es que, probablemente, nunca tenga que utilizar funciones trigonométricas en un programa en C/M. Esperemos, no obstante, que la rutina le sirva de ejemplo sobre cómo manejar el calculador.

Brother EP-44

Tengo un Spectrum con el Interface-1 y cable RS-232 de Investrónica, y por otro lado, una Brother EP-44, con la que os escribo. El caso es que para enviar textos no hay ningún problema, pero al intentar pasar un listado, los GDU me salen como un signo de interrogación. Me gustaría saber si esto se podría solucionar, aunque sea con un programilla.

José L. GRAU - Madrid

☐La Brother EP-44 es una máquina de escribir, no una impresora, por lo que carece de posibilidades gráficas. Sólo le admitirá caracteres ASCII, es decir, aquéllos con códigos comprendidos entre 32 y 127 ambos inclusive. Los GDU tienen códigos superiores a 127, por lo que aparecen como un signo de interrogación.

En las impresoras matriciales normales, la solución está en listar por pantalla y hacer un «COPY» (es el procedimiento que empleamos nosotros para los listados que publicamos), pero la EP-44 no admite «COPY», ya que no trabaja en modo gráfico. Ciertamente, creemos que el problema no tiene solución.

Programación en C/M

Quiero preguntar cómo se hace para poner un programa en código máquina, porque cuando lo tecleo, la primera letra no aparece, y aparece, en su lugar, el comando de la tecla.

Jordi GUAL - Barcelona

□Su duda es muy frecuente entre los principiantes, y no es ésta la primera carta que recibimos con la misma pregunta.

El Spectrum lleva incorporado un intérprete de lenguaje Basic. Cuando se conecta el ordenador, el usuario se encuentra con el editor de Basic, por lo que sólo puede teclear comandos de este lenguaje. Para programar en código máquina, lo más cómodo es escribir el programa en Assembler y, luego ensamblarlo. El Spectrum no permite escribir directamente en Assembler, por lo que no hay más remedio que utilizar un programa denominado «Ensamblador». Estos programas llevan incorporado un editor de Assembler que le permitirá teclear su programa con toda facilidad.

Primeros pasos

He estudiado el programa de instrucciones que viene con el ordenador Spectrum 48 K y veo que me faltan aún muchos conocimientos para poder entender los programas que editáis en la revista. Por ello, os pido que me aconsejéis qué libro o lo que puedo estudiar para entender dichos programas.

Juan J. LOBERA - Pamplona

□Por su carta, deducimos que acaba usted de entrar en este maravilloso mundo de la Informática. No se preocupe si le cuesta empezar a entenderlo, nadie nace sabiendo pero, sobre todo, no se desanime. Existen un montón de libros y publicaciones para ayudarle y, por supuesto, puede recurrir a nosotros para todo lo que necesite.

Los programas que editamos en nuestra revista están en Basic o código máquina. Le recomendamos que empiece por el primero que es más fácil. Para ello, puede servirle el propio «Microbasic» de Rafael Prades que puede solicitar por correo a nuestra editorial. Si le apasiona el tema (de lo que estamos seguros) y decide continuar con el código máquina, le recomendamos el Curso de Código Máquina publicado en las páginas centrales de MICROHOBBY en los números 42 al 95. No obstante, empiece por el principio y vaya avanzando poco a poco. En menos de un año puede dominar ambos lenguajes.

Monitores

Tengo la posibilidad de adquirir, a muy buen precio, un monitor de vídeo (fósforo verde), procedente de un circuito cerrado de televisión de unas oficinas. Pero mi problema es que, habiendo consultado con la delegación de Investrónica de mi ciudad, en cuanto a si me serviría para mi Spectrum, la única respuesta obtenida ha sido que ellos me instalaban una salida de vídeo compuesto, para monitor, sin saber decirme si esto sería compatible con este tipo de monitor. Por cierto que mostraron muy poco interés por mi consulta.

He de deciros que yo lo he enchufado a la salida de vídeo de mi vídeo-cassette, para ver si era compatible con esa salida y no he logrado siquiera que se ilumine la pantalla.

Jorge J. ARDIT - Oviedo

□Es lógico que no hayan podido darle una respuesta concisa sin ver y probar el monitor. La mayoría de estos monitores tienen entrada de vídeo compuesto, pero puede ocurrir que el suyo utilice un sistema distinto. En cualquier caso, cualquier monitor que pueda ser compatible con el Spectrum, también lo debe ser con la salida de vídeo compuesto de un vídeo-cassette. Compruebe, no obstante, si el problema es que el monitor está averiado.

Respecto a instalar la salida de vídeo compuesto en el Spectrum, es tan fácil que vale la pena hacerlo uno mismo. La señal se toma del más exterior de los dos conductores que entran al modulador y se aplica a la pata central de un conector hembra tipo «RCA» a través de una resistencia de 10 ohmios (para evitar averías en caso de cortocircuito). La masa puede tomarse del cuerpo metálico del modulador.

Spectrum Plus II

He leído que han sacado el Spectrum Plus II, hecho por Amstrad, por lo que me interesaría saber si van a dejar de fabricar el Spectrum Plus.

En caso afirmativo, ¿podría cambiar el Spectrum Plus por un Plus II dando la diferencia, o tendría que comprar uno nuevo?

Alexis RODRIGUEZ - Cádiz

□La aparición en el mercado del Spectrum Plus II hace pensar que el modelo anterior dejará de fabricarse y, de hecho, es prácticamente seguro que así sea. Tenga en cuenta que el Plus II sale al mercado casi al mismo precio que tenía el antiguo Plus.

Respecto a la posibilidad de cambiar uno por otro, no creemos que sea posible, dado que se trata de dos ordenadores bastante diferentes, a pesar de ser compatibles. Cuando Investrónica dio la oportunidad de cambiar el 48 K por el Plus no se trataba, realmente, de un cambio. Lo que hacían en realidad era ponerle un 48 K al teclado de un Plus. En este caso, no es posible realizar la misma operación, ya que el Plus II es completamente diferente del modelo anterior.

DE OCASION

● VENDO Impresora matricial Seikosha GP-50S (interface directo para Spectrum), en perfecto estado, por el precio de 15.000 ptas. Interesados dirigirse a: Daniel Vidal Llach. Conde Salvatierra, 28. 80225 Terrasa (Barcelona).

● QUIERO vender el siguiente lote y a un precio de 33.000 ptas.: Zx Spectrum 48 K, con interface Kempston, joystick Quick Shot I, libro de instrucciones en castellano y cinta horizontes. El ordenador sólo por 28.000 ptas. Interesados en la compra pueden llamar al tel. (93) 235 06 74 y preguntar por Juan (hijo) o bien escribir a la siguiente dirección: Juan Lesán. Centro, 6-8, entlo 1.°. 08026 Barcelona.

● VENDO consola de videojuegos CBS-Colecovisión con 6 cartuchos y volante (todo con instrucciones en español y cables) por el precio de 30.000 ptas. Interesados escribir a Ignacio Carrera. Manuel de Castro, 8, 9.° Vigo (Pontevedra).

● DESEARIA contactar con usuarios de Spectrum 16/48 K para el intercambio de pokes, trucos, ideas, mapas, esquemas. Interesados escribir a Manuel Rodríguez. Cortijo, 12, 5.°. 33012 Gijón. Tel. (985) 32 61 32.

● VENDO Spectrum 48 K, con un interface tipo Kemspton, todos los cables y conexiones, instrucciones en inglés con la cinta de demostración, una funda, revistas. Todo por tan sólo 17.500 ptas. Llamar al tel. 668 26 35 de Barcelona.

● CLUB de usuarios de Spectrum 16/48 K, para toda España. Intercambiamos, trucos, ideas, mapas, etc. Interesados escribir a la siguiente dirección: Juan Ramón García Vila. Fontela s/n. Burela (Lugo). O bien llamar al tel. (982) 58 10 81.

● ME GUSTARIA intercambiar trucos, rutinas, listados, pokes, etc. Interesados dirigirse a: José García Agustín. Maestro Serrano, 19. 46830 Benigamin (Valencia).

● CAMBIO videojuegos Sistem Atari CX/2600 con joystick. Además regalo un amplio catálogo de la casa Atari con 70 cartuchos para pedir y seis posibilidades distintas de mandos. Todo esto lo cambio por un Spectrum 16/48 K aunque tenga el teclado un poco estropeado, o en su defecto, un Oric, VIC 20 o un Hit Bit 55p. Aceptaría otras ofertas. Interesados llamar al tel. (927) 53 19 85 o bien escribir a J. C. Ardilla. Góngora, 3, 3.°. Navalmoral de la Mata (Cáceres).

● VENDO ordenador Spectrum 48 K con manuales, cintas de demostración, funda, joystick, interface tipo Kempston y 20 revistas por sólo 38.000 ptas. (negociables). Regalo libro de C/M valorado en 2.200 ptas. Escribir a David García Cobrí. Oca, 73, 4.° B. Madrid.

● VENDO Spectrum Plus, comprado en abril-85, pero con muy poco uso, con todos sus complementos. Cassette Computone comprado en agosto-86. 90 números de la revista MICROHOBBY Y 14 de Micromanía. Precio total 50.000 ptas. (se regala curso completo de Basic en cassette valorado en 12.000 ptas.). También vendo órgano Casio CT-600, con embalaje original, parrilla portapartituras. Precio 70.000 ptas. Además NCRCM-V, 2 unidades de disco 5 1/4 con teclado profesional, impresora Citoh, 120 c/p. Precio 400.000 ptas. (usado en videoclub). Interesados contactar con José A. Rodríguez, llamando al tel. (952) 39 20 53 de Málaga de 10 a 2 y 5 a 10.

● VENDO interface programable con amplificador de sonidos, bus trasero adiccional y led indicador on/off. Muy fácil de usar y no necesita software. Para más información llamar al tel. 201 67 36 de Madrid y preguntar por José Vicente.

● VENDO 80 revistas de MICROHOBBY además de estar interesado en formar un club de usuarios en la zona de Getafe de Madrid. Llamar por las mañanas y preguntar por Javier al tel. (91) 696 14 03 o bien escribir a la siguiente dirección: Javier Agudo Mendoza. Valencia, 45, 1.° A. Getafe (Madrid).

● ATENCION compro números atrasados de MICROHOBBY. También fotocopias de las instrucciones y de los mapas de los siguientes juegos: Zorro, Fairlight, Camelot Warriors, Nodes of Yesod, Tomahowk. Los gastos de envío y las fotocopias a mi cuenta. Interesados llamar al tel. (96) 138 49 31 y preguntar por Ramón.

● NECESITO las instrucciones en castellano del juego Flight Simulation de la casa Psion. Pagaría las fotocopias. Interesados dirigirse al tel. (926) 42 48 90 a partir de las 8,30 de la tarde. Preguntar por Diego.

● VENDO Spectrum 48 K con todos sus accesorios. Regalo revistas de Input y ZX. Buen precio. Para localización, llamar al tel. (91) 246 57 54. Preguntar por Juan.

● COMPRARIA las fotocopias de las instrucciones de los siguientes programas: Saboteur, Fairlight y Finders Keepers. Precio a convenir. Interesados llamar a Ismael Muñoz Sánchez. Avda. de Madrid, 60-A, 5.° A. 23008 Jaén.

● BUSCO las instrucciones de los programas Super-Chess 3-5 y Tape Copier 7. Las cambiaría por trucos, mapas pokes. Interesados escribir a la siguiente dirección: Rubén Domingo Herrero. Camino Leganés, 37. Móstoles (Madrid).

● VENDO ZX Spectrum 48 K, con libros, cinta de demostración, fuente de alimentación, cables, cassette especial para el ordenador, interface tipo Kempston, joystick Quick Shot II, amplificador de sonido, psicodélico de tres canales, llegando hasta 1.000 w con bombillas. Todo ello por sólo 23.000 ptas. Ponerse en contacto con Vicente al tel. 465 61 04 de Madrid.

● DESEARIA contactar con usuarios del Spectrum 48 K, para intercambiar ideas, trucos, información. Interesados dirigirse a: David García García. Rey, 37, 2.° E. 28300 Aranjuez (Madrid). Tel.(91) 891 15 16.

● REGALO, la cinta Chess (ajedrez) o Survival a quien me mande las instrucciones completas del programa Pyjamarama. Interesados escribir a Emilio Fernández. Manuel Calvo, 30, 3.° Iz. Portugalete (Vizcaya).

● VENDO ordenador Spectravídeo 328-80 K Ram con cassette SVI-904 por 18.000 ptas. Llamar a Daniel Majoral Marco. Consejo de Ciento, 384, 5.ª, 1.°. 08009 Barcelona. Tel. (93) 245 19 84.

● PARA Spectrum, vendo lo siguiente: interface joystick programable «Indescomp» (3.500) Mega-Sound «Cheetah» (2.000); Conmutador TV-ordenador (500); 13 primeros números de «Todospectrum» (1.500); 17 «ZX Spectrum» (1.500); colección completa de «Circuito Impreso» (48 números, 8 años, 12 primeros encuadernados) (3.000); los 47 primeros de «Elektor» (5.000 ptas.); 10 de «Nueva Electrónica» (1.000), 80 número de la revista MICROHOBBY semanal más estuche incluido (5.000). Interesados llamar al tel. (91) 246 39 30. Fernando (hijo). Noches.

● ¿QUIERES gozar de buena música y de una imagen nítida en tu Spectrum? Vendo: sintetizador musical (Ay-3-8910): 3 canales, 8 octavas, generador de envolventes y ruido, salida amplificador, amplificador de vídeo para monitores monócromos y color. Si te interesa llama al tel. (93) 334 47 62 y pregunta por Javier.

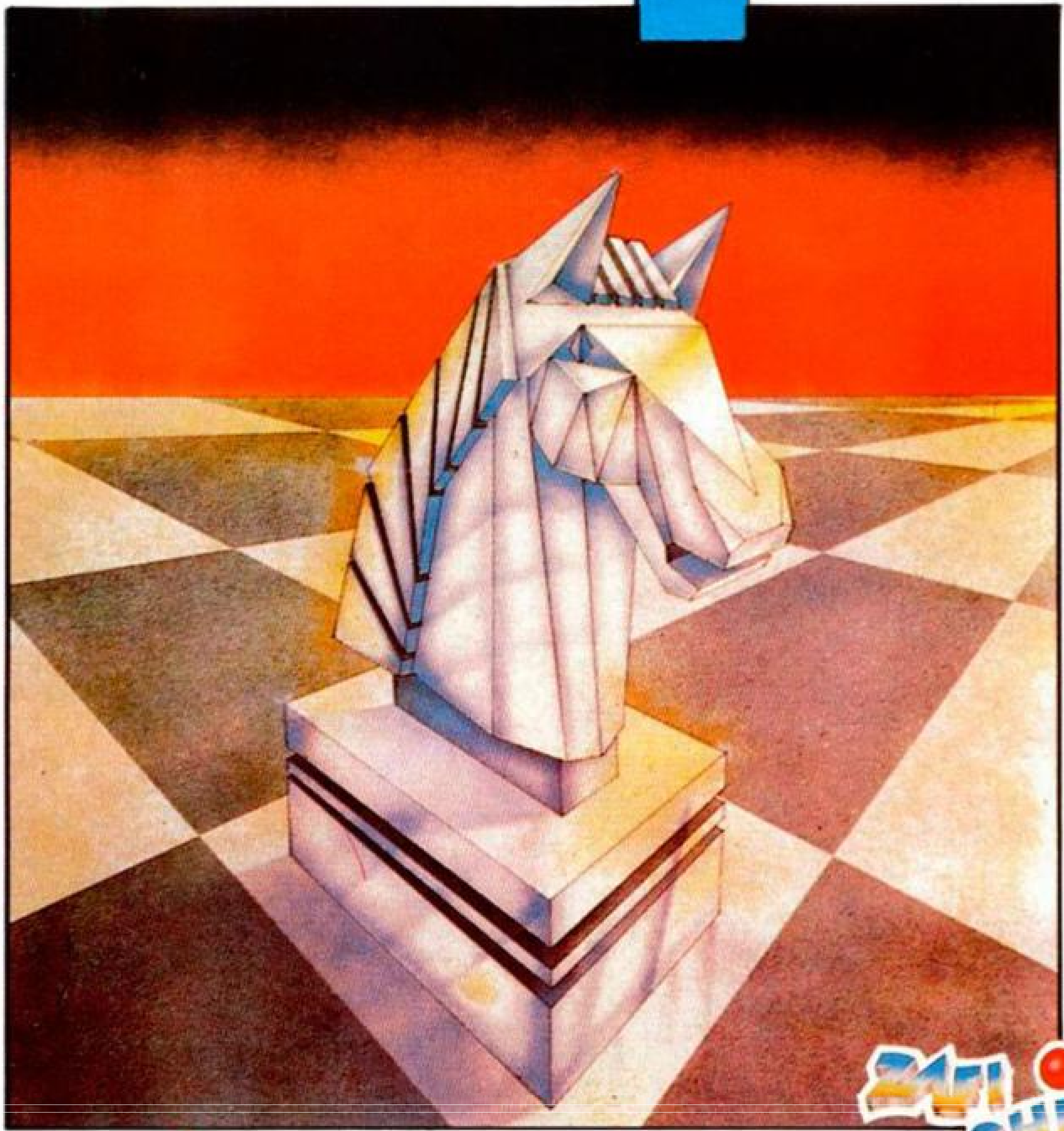
ATENCION

Micromanía

Necesita colaboradores con conocimientos de Código Máquina de MSX, que además sean viciosos en el arte del dominio de los juegos. Los interesados enviar los siguientes datos:

- **Nombre.**
- **Dirección.**
- **Provincia.**
- **Teléfono si se tiene.**
- **Conocimientos que se posean.**

COLOSSUS 4 CHESS



Spectrum 48/128K
Commodore
Apple II/II+/IIc/IIe

- Imagen 3-D
- Control opcional de joystick

Colossus Chess, vencedor frente a todos los programas de ajedrez conocidos.

El programa de ajedrez más avanzado del mundo.

¡Rétale!

ZAFIRO SOFTWARE DIVISION
Paseo de la Castellana, 141. 28046 Madrid.
Tel. 459 30 04. Tel. Barna. 209 33 65.
Telex: 22690 ZAFIR E

IMPRIME MOVIEGRAF. C/. ESTIGIA, 3 - 28037 MADRID